Decentralizing the Web: A Study Of The Interplanetary File system

Matthew Wanner

Why I choose this topic

- I wondered why in old online games there would be a host migration
- A lot of video games today use a Client-Server architecture to host online games
- Old online games would use a peer-to-peer (P2P) hosted meaning one of the players systems was hosting the lobby
- I thought it was cool that they didn't rely on a central server
- I learned what P2P networks were and thought they were interesting and wanted to learn more
- Did some research and found study’s on the Interplanetary File System (IPFS) a popular P2P network

Outline

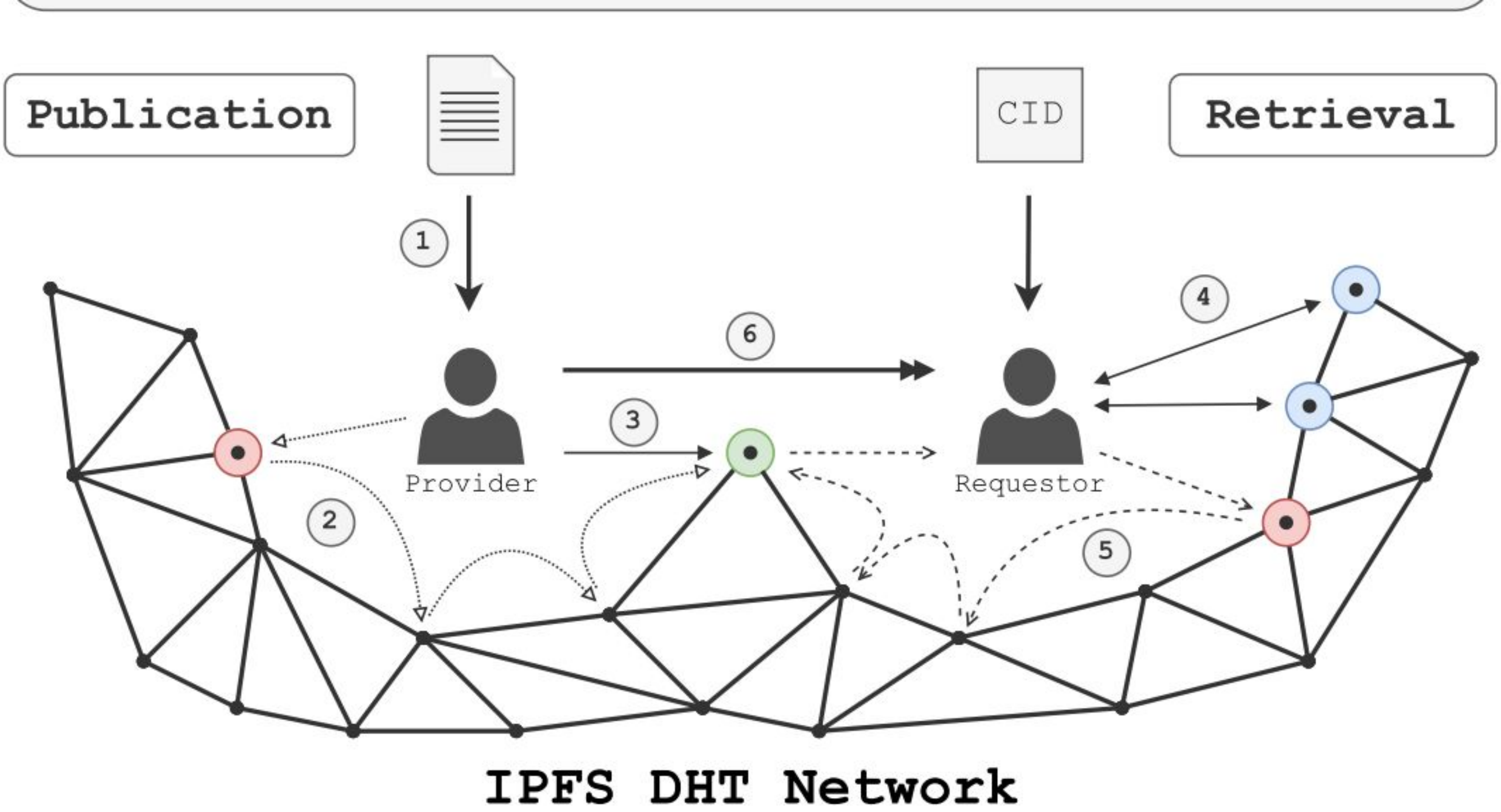
- IPFS overview
- Analysis of IPFS performance
- Use cases and application of IPFS
- Analysis of IPFS use in restricted environments.
- Conclusion
- Q&A

IPFS Basics

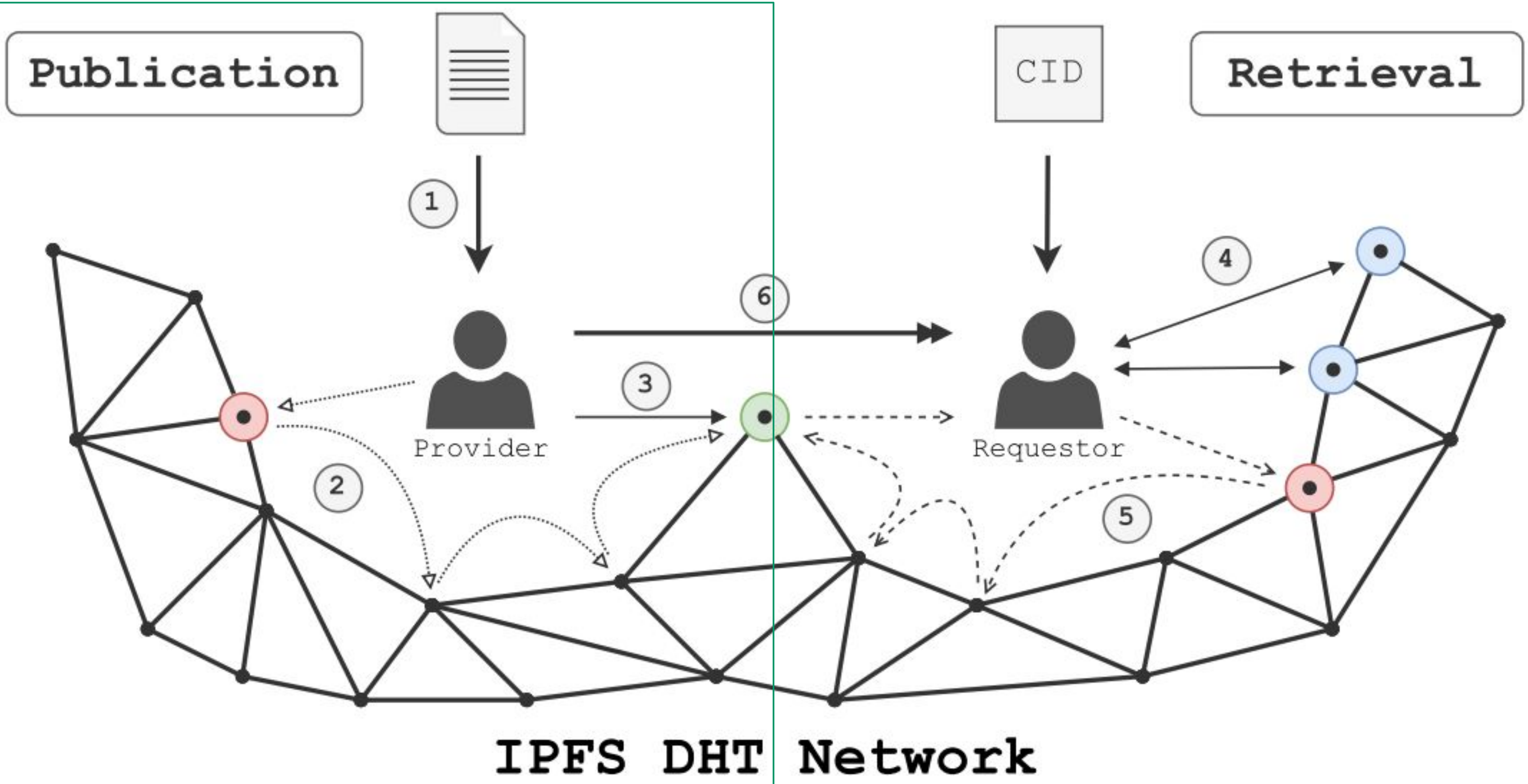
- Traditional internet:
  - Relies on a client-server model
  - Clients request resources from a central server
  - Files located and accessed via URLs that point to their location on a server
- IPFS network:
  - Operate through content addressing
    - Uses unique content identifiers (CIDs) to access files based on their content, not location
  - Each peer on IPFS stores and serves content, contributing to the network

Why use IPFS?

- Benefits of IPFS:
  - Independant
    - Moves away from reliance on central servers by distributing data across multiple nodes
  - Resilient
    - Network that can withstands outages and censorship more effectively than centralized models
  - Data permanence
    - Content remains accessible as long as its hosted somewhere on the network



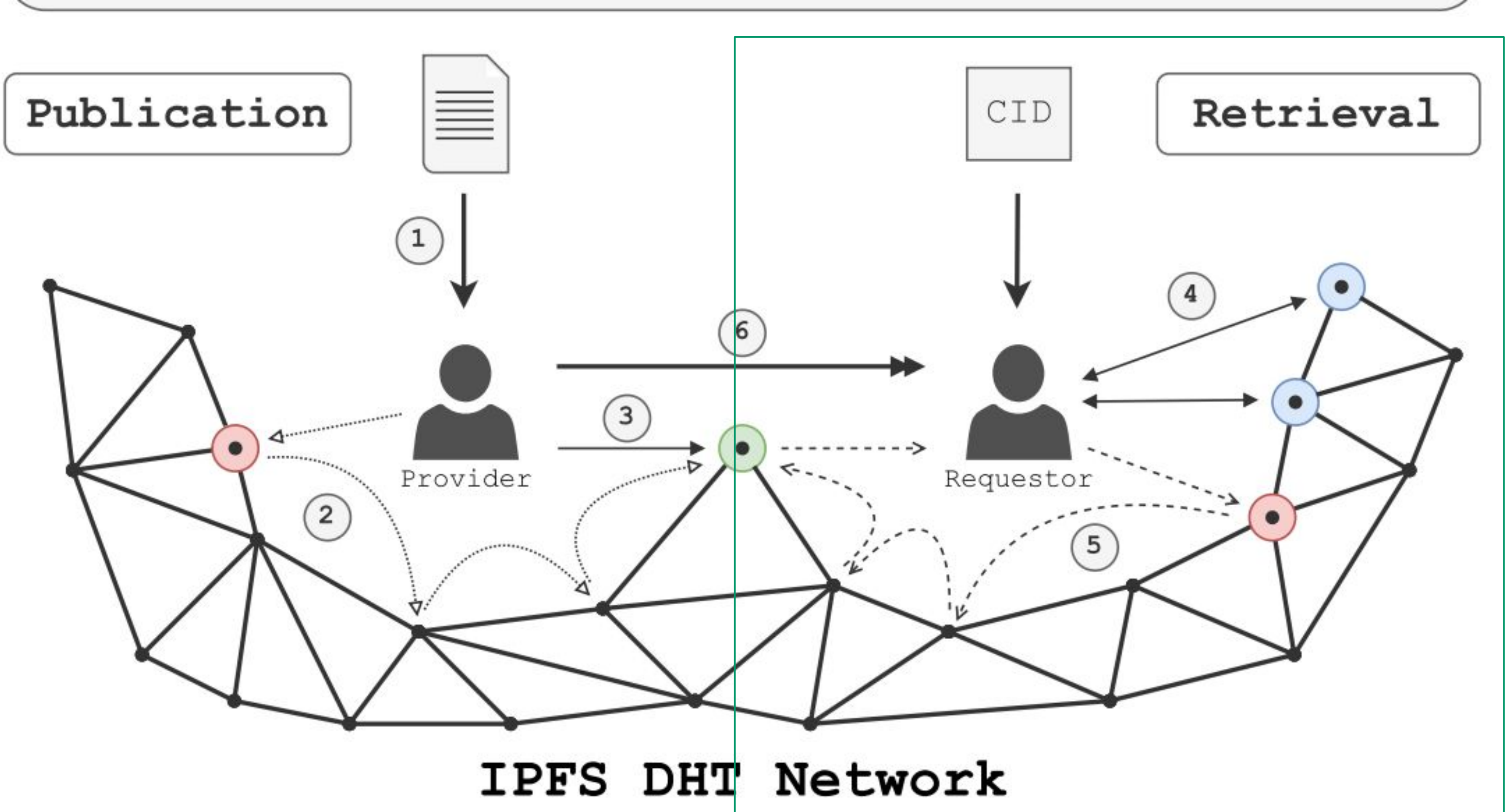
Publication
CID
Retrieval
1
2
3
4
5
6
Provider
Requestor
IPFS DHT Network



Publication
Retrieval
CID
1
2
3
4
5
6
Provider
Requestor
IPFS DHT Network

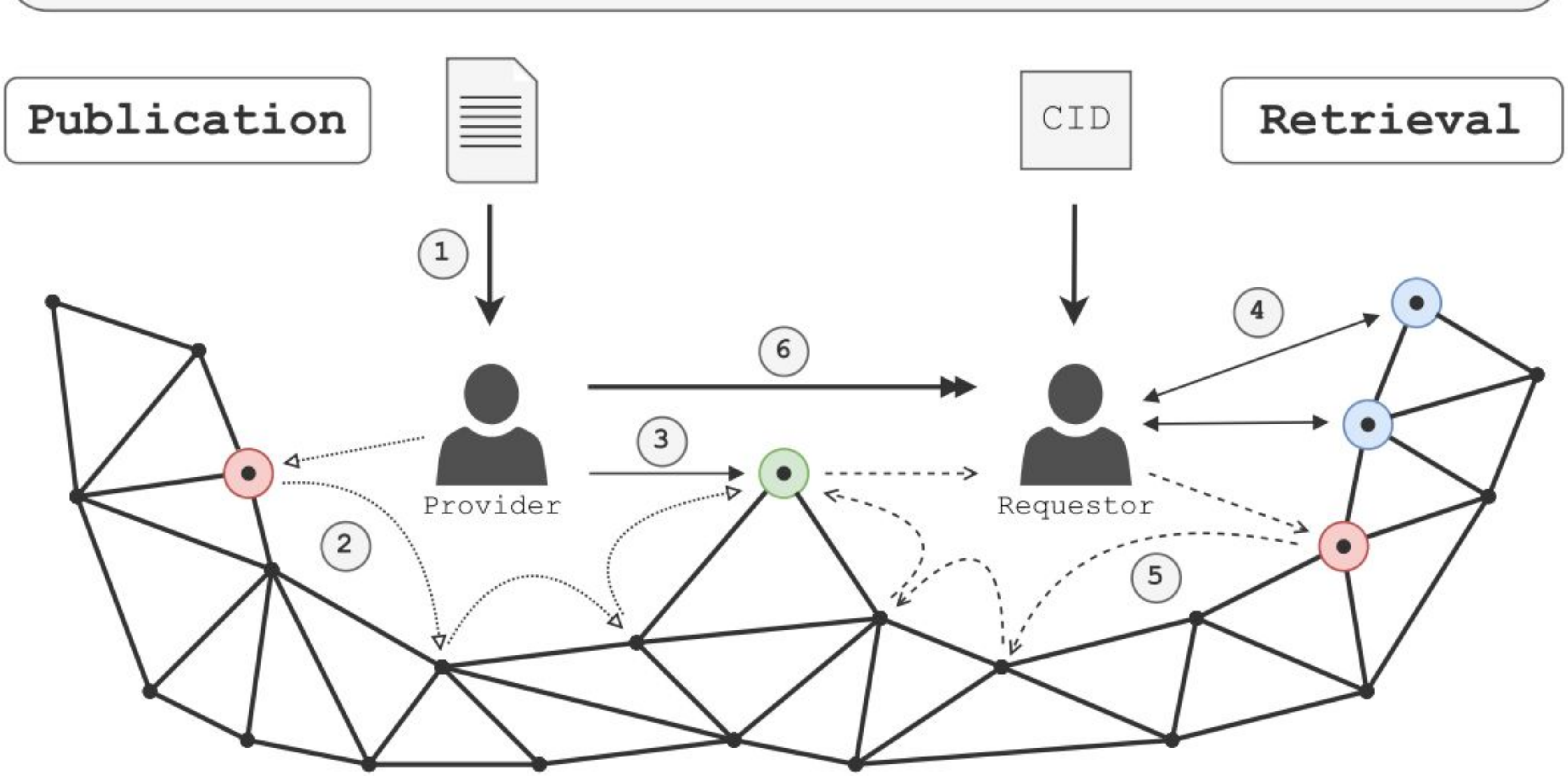
Adding a File to IPFS

- Every participant in the network acts as a node, contributing to the storage and retrieval of files
- When a file is added to the IPFS from someones computer (node) it is hashed to generate a Unique Content Identifier (CID)
- Node creates a provider record saying to the network that it has the file associated with the CID
- Distributed Hash Table (DHT) gets updated with the provider record
  - DHT is a decentralized directory that maps files CIDs to nodes that hold them
- Provider record is stored on the 20 “closest” nodes to the CID based on the DHTs distance metric


Publication
Retrieval
CID
1
2
3
4
5
6
Provider
Requestor
IPFS DHT Network

Retrieving a File from IPFS

- Look up file by its CID
- When a node wants to retrieve a file, it queries the DHT with the CID of the file it wants
- The DHT responds with the provider records for that CID
- Querying node uses information from provider record to connect to provider node
- Querying node receives file from provider node
- Local node verifies file by hashing it and comparing the hash to the CID to ensure integrity
- This node can now act as a provider of the file to other nodes. Enhancing availability across the network



IPFS DHT Network

How Well Does it Work? Analysis of IPFS data

- Go through a study that covers
  - Performance Data
  - IPFS Gateway usage Data
  - Peer Data

Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web

Authors: Trautwein, Dennis and Raman, Aravindh and Tyson, Gareth and Castro, Ignacio and Scott, Will and Schubotz, Moritz and Gipp, Bela and Psaras, Yiannis

Publisher: Association for Computing Machinery

Performance Data Collection Methodology

- 6 virtual machines across 6 regions to simulate IPFS nodes
- Each machine ran an IPFS node to conduct controlled tests to assess how efficiently nodes can publish and retrieve content
- A node would announce a new 0.5 MB object to the network
- The other nodes then attempted to locate and download that object over the network
- After retrieval, nodes disconnected to ensure a fresh test environment for the next run
- Number of successful publications and retrievals were recorded

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

Analysis of Performance Data: Publication

- Median publication time across regions is 33.8 seconds
- 90th and 95th percentile times are 112.3 and 138.1 second respectively
- Delays are consistent across regions
- DHT walk is primary contributor to publication delay (87.9% on average)
- Improving DHT walk efficiency is a key area for future improvements

Analysis of Performance Data: Retrieval

- Achieved a 100% success rate
- Performance overview
  - Noticeable variability in retrieval times
  - On average, retrievals take longer than loading a typical web page but are faster than content publications on IPFS
- Retrieval speed
  - 2.9 seconds for 50th percentile (median)
  - 4.34 seconds 90th percentile
  - 4.74 seconds 95th percentile
- Regional Variations
  - Central Europe fastest median time at 1.81 seconds
  - South Africa slowest median time at 3.75 seconds
- Reason for efficiency in retrieval vs publication
  - Publication DHT walks need to find 20 nodes while a retrieval walk ends upon finding a single node.

AWS Region	Publication Percentiles 50th	90th	95th	Retrieval Percentiles 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

Af_south_1 = Cape Town

Ap_southeast_2 = Sydney

Eu_central_1 = Frankfurt

me_south_1 = Bahrain

sa_east_1 = Sao Paulo

us_west_1 = N. California

IPFS Gateway data Collection Methodology

- Gateways are a different way to interact with IPFS
  - 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
- Authors collected and analyzed GET requests from a public IPFS gateway
- Focuses on traffic from one day in January 2022 at a gateway located in the US
- Examined 7.1 million requests looking at details like
  - When request was made
  - What kind of device was used
  - Where the request came from
  - Volume of data transferred per request
  - Cache hit rate

Analysis of Gateway Data

- User Engagement
  - Identified 101,000 users
  - Accessing 274,000 unique CIDs
- Data size
  - Average size of requests was 664.59 KB
  - 79.1% of requests were larger than 100 KB
  - No correlation between object size and latency, suggest other factors affect delay
- Speed and efficiency
  - 46% of requests were fetched instantly, indicating a cache hit
  - Most remaining requests served under 24ms
- Over half of traffic (51.8%) of traffic came from third party sites
  - Significant portion of referred traffic from a small number of sites, mainly streaming and NFT platforms

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

Nginx is a cache for HTTP requests, caching responses to speed up response times

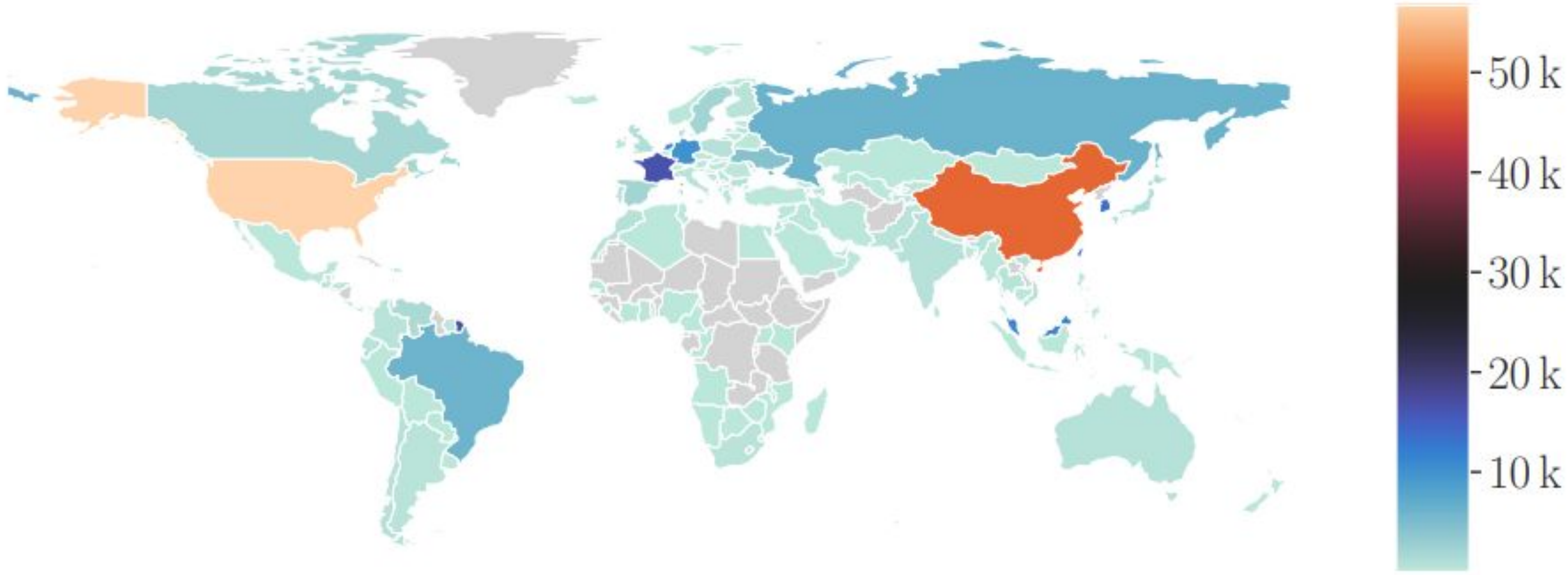
Authors note that while gateways can centralize certain aspects of IPFS, the possibility for anyone to set up a gateway helps maintain the decentralized ethos of IPFS

Peer Data Collection Methodology

- Researchers utilized a crawler to collect peer data due to a lack of a centralized peer directory.
  - A crawler is a software tool to traverse and gather data about networks.
- Operated crawler from a server in Germany every 30 minutes.
- Crawler systematically queried nodes starting with 6 established IPFS nodes and expanded outwards until no new peers were found.
- Crawler ran over 9,500 times and compiled a detailed list of peers including location, how long the peers stayed online, and technical details

Analysis of Peer Data

- Identified 198,964 peers across 152 countries
- Reachability
  - 54.5% of IPs could be reached at least once
  - 45.5% were never reachable
- Geographical Concentration
  - Highest concentrations of users were in the US (28.5%) and china (24.2%)
  - France, Taiwan, and South Korea were next highest
- Reliability
  - Only 1.4% (2,747) of peers showed >90% uptime, considered “reliable”
  - About on-third of peers were never accessible (highlights network resilience)
- Distribution of users ensures no single country can dominate or disrupt IPFS, maintaining decentralization
- A concerning finding is that the top 10 IP addresses host nearly 66k distinct PeerIDs, raising concerns about misuse and its impact on routing


50 k
40 k
30 k
20 k
10 k

Use Cases of IPFS

- Video on demand
- File sharing
- Social networking services
- NFTs (Non-Fungible Token)

Dude, where's my NFT? Distributed Infrastructure for digital Art

Authors: Leonhard Balduf, Martin Florian, Björn Scheuermann

IPFS and NFTs

- Digital Art
  - NFTs typically represent digital assets like art, music, videos, or other creative work
- Blockchain storage
  - Typically the blockchain stores the NFT (another P2P network)
  - Blockchain is expensive and inefficient

Role of IPFS to NFTs

- Provides a decentralized solution for NFTs, overcoming limitations of traditional blockchain storage
- Utilizes content-addressing capabilities, ensuring data immutability and authenticity
- Enhances accessibility and reliability, as files are redundantly hosted across multiple nodes
- IPFS reduces costs associated with data storage on blockchain

Note: I'm not an NFT enthusiast. I think this highlights the benefits and versatility of IPFS

I’m InterPlanetary, Get Me Out of Here! Accessing IPFS From Restrictive Environments

Authors: Leohard Balduf, Sebastian Rust, Björn Scheuermann

Assessing IPFS's functionality in restricted environments

- Testing China's Great Firewall (GFW)
- Researches experiment setup
  - 2 non-NATed (Network Address Translation) machines set up (controls)
    - A NAT provides a layer of security by hiding IP addresses from external devices
    - One in Germany and another in the US
  - 2 NATed machines set up
    - One in China
    - One in US act as a control for the NATed machine in China to measure the additional impact of the GFW
- Tested IPFS's locally hosted node data exchange and gateway accessibility

Overcoming Censorship: Gateway testing

- Tested 81 public gateways listed by the IPFS community
- All gateways were hosted outside of China, necessitating data to traverse the GFW
- Attempt to retrieve a widely replicated text file through the public IPFS gateways from each node
- Check for any use of whitelisting or other selective content delivery mechanism by verifying its hash

Gateway connectivity Results

- 14 of the 81 gateways worked correctly from the non-NATed clients in germany and US
    - Author notes that the gateway list was community maintained and may have outdated entries
- One of those 14 gateways was inconsistently accessible from US NATed node, suggesting flakiness
- Only 5 gateways were functional from the node in china, indicating challenges but not complete blockage by the GFW

Gateway Connectivity Results

(a) Functionality by Vantage Point.

Machine	Tested	Working
DE Client (non-NATed)	81	14
CN Client	81	5
US Client	81	14
US Client (non-NATed)	81	13

IPFS Client Node Testing

- Using the same 4 vantage points. This time using IPFS client software instead of gateways
- Create random files to ensure they're the only ones providing that content
- Each node downloads content from a different node in a random order in rounds
- Conduct test over 7 days, amounting to about 2000 data points per vantage point
- Verified hash for integrity of downloaded content
- Authors note that downloading and setting up an IPFS Client in China was more difficult but not insurmountable (Due to some sources of downloading the software being blocked)

Overcoming Censorship: IPFS Client Testing Results.

- Attempted 8,064 downloads across the 4 nodes with a 71% success rate
- Downloading
  - German non-NATed client success rate = 58%
  - US non-NATed client success rate = 66%
  - US and China NATed clients had success rates of 80%
- Uploading
  - German and US non-NATed clients success rates were >90%
  - US and China NATed clients success rates both were ≈ 50%
- This shows that IPFS client nodes are functional even in restricted environments
- Authors note that in a real world scenario, a node would likely download from multiple nodes, increasing chances for success

Nature of NATs on P2P networks.

- NATed Networks
  - Downloads are easier from both NATed and non-NATed networks
  - Uploading is harder especially to non-NATed networks. Uploading to other NATed networks is more successful
- Non-NATed Networks
  - Downloading is harder if downloading from a NATed network
  - Uploading is easier because these networks are directly accessible

Overcoming Censorship: IPFS Client Results

Download Success Rate by Storing Machine

Stored On	n	Succ.	Rate
DE Client (non-NATed)	2016	1834	0.91
CN Client	2016	1049	0.52
US Client	2016	956	0.47
US Client (non-NATed)	2016	1873	0.93

Download Success Rate by Downloading Machine

Downloaded By	n	Succ.	Rate
DE Client (non-NATed)	2016	1160	0.58
CN Client	2016	1621	0.80
US Client	2016	1608	0.80
US Client (non-NATed)	2016	1323	0.66

Overcoming Censorship Conclusion

- Functional
    - IPFS operates effectively even in restrictive environments, overcoming barriers to information sharing
- Points of Failure
    - Public gateway access could be limited
    - Distribution of IPFS software
- As long as these risks can be mitigated, IPFS can serve as a powerful tool for sharing information freely in environments where access is often constricted

Conclusion

While IPFS or other P2P network may never surpass the performance or replace the traditional client-server architecture, the various studies that I've talked about today shows 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.

Q&A: