

Problems

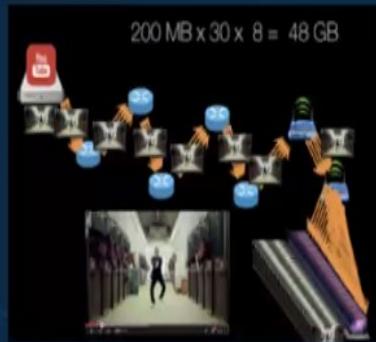


emerging networks



censorship

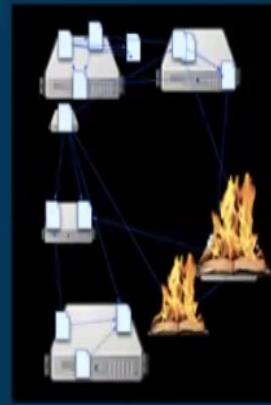
Addresses



huge inefficiency



bad security model



links break



no offline use



4,000+ IPFS Community Contributors

Content Routing Speed

IPFS 0.4.23 vs IPFS 0.5.0

Avg **8 seconds**

1.5 seconds

95th % **42 seconds**

14-20 seconds

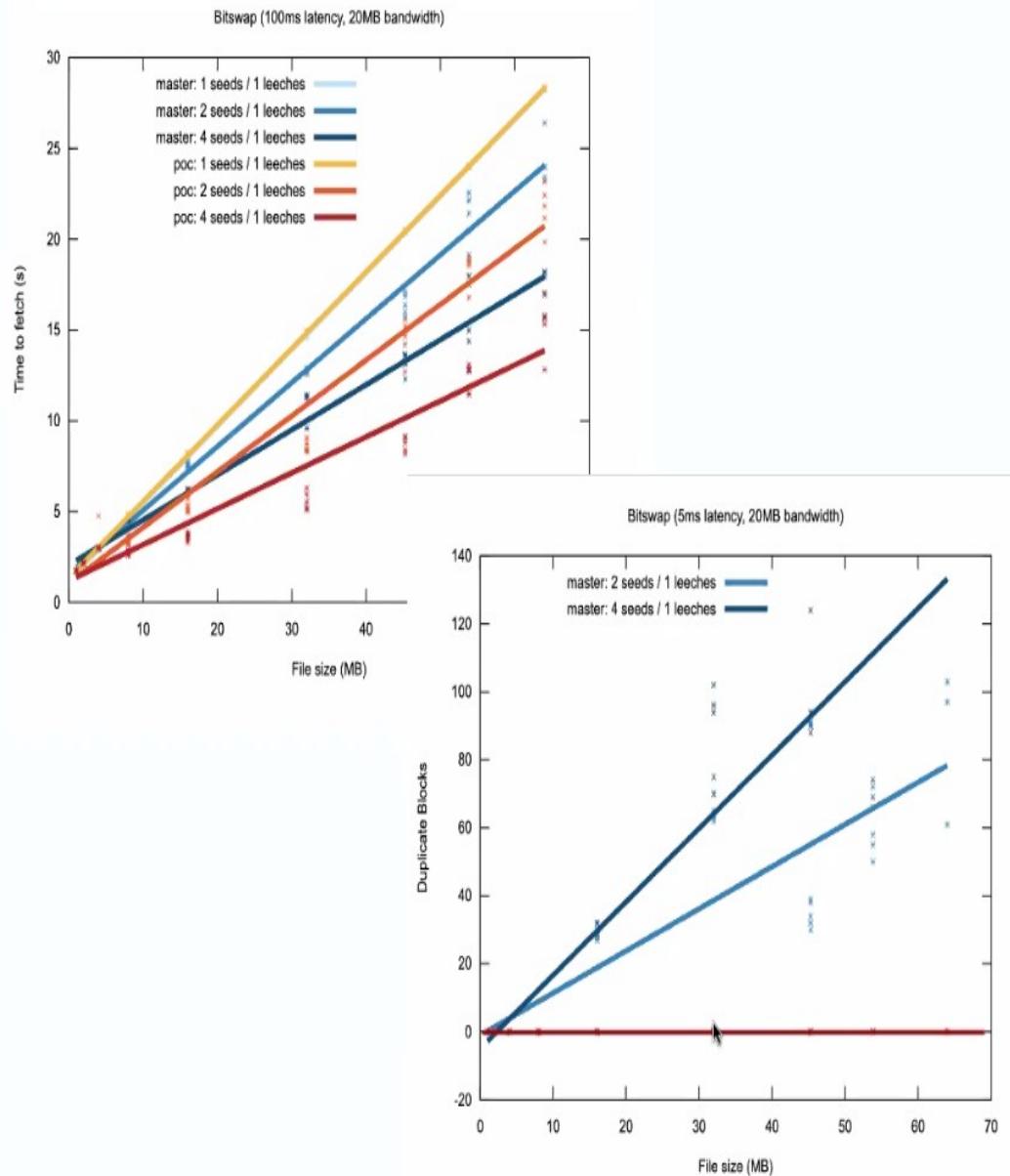
Content Routing: Public Network

- Numbers:
 - 50%: 8s → 1.5s
 - 95%: 42s → 14-20s
- Notes:
 - These numbers are highly variable.
 - They'll get better as the network upgrades.
 - More improvements on the way.



Content Exchange: Bitswap

- Substantial protocol changes to:
 - Reduce wasted bandwidth due to duplicate blocks.
 - Increase parallelism and throughput..



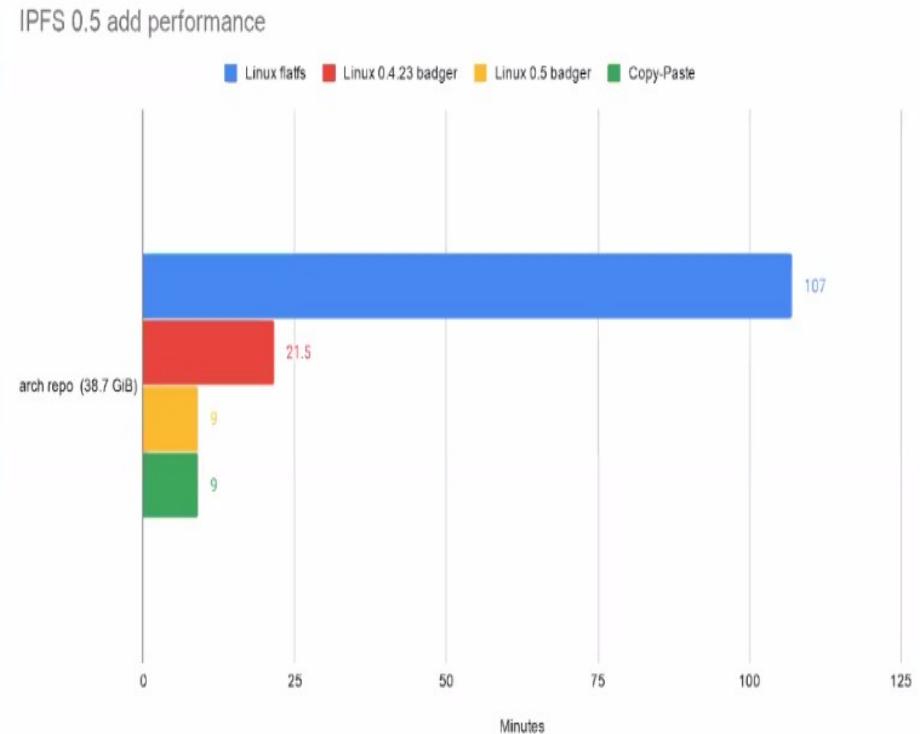
Content Exchange: Graphsync

- Go-ipfs 0.5 can respond to graphsync requests (when the experiment is enabled).
- What is Graphsync?
 - Successor to bitswap.
 - Syncs at the IPLD graph layer instead of at the block layer.
 - Can request an entire graph all at once (e.g., an entire file or directory tree).
- What's missing?
 - Requesting data via graphsync.
 - Parallelizing graphsync requests.
 - All the "smarts" that makes bitswap "just work".



Import/Export: Add Performance

- Async datastore writes in badger with explicit sync points.
- 2-3x performance improvement over badger with sync writes.
- Performance on-par with copy/paste.
- Badger still not enabled by default due to two outstanding issues:
 - 1GiB Memory Spikes: dgraph-io/badger#1292
 - GC not effective: ipfs/go-ds-badger#54



Import/Export: Dag Import/Export

```
~$ ipfs dag export QmQPeNsJPyVWPFDVHb77w8G42Fvo15z4bG2X8D2GhfbSXc \
| xz > welcome_to_ipfs.car.xz
```

```
0s 6.73 KiB / ? [-----] 5.16 MiB/s 0s
```

```
~$ xz -dc welcome_to_ipfs.car.xz | ipfs dag import
```

```
Pinned root    QmQPeNsJPyVWPFDVHb77w8G42Fvo15z4bG2X8D2GhfbSXc    success
```

```
~$ ipfs dag export QmQPeNsJPyVWPFDVHb77w8G42Fvo15z4bG2X8D2GhfbSXc | xz | nc ...
```

Libp2p: Improved AutoNAT

- In 0.5
 - Faster NAT detection.
 - Rate limiting.
 - Dial-back filtering and restrictions.
 - AutoNAT service enabled by default on all go-ipfs nodes (highly rate-limited).
- Next Steps
 - Even faster NAT detection in 0.6
 - Deeper integration into libp2p to avoid advertising undialable addresses.



Libp2p: TLS By Default

- TLS is now the default transport.
- Well understood security properties.
- Faster.
- Next up:
 - Noise js interop.
 - Secio deprecation in 0.7.0 likely.



Libp2p: QUIC Draft 27

- Final QUIC protocol version before we stabilize it.
- Will be enabled by default in 0.6



QUIC

Libp2p:

P2P Address Format

/ip4/104.131.131.82/tcp/4001/**ipfs**/QmaCpDMGvV2BGHeYERUEnRQAwe3N8SzbUtfsmvsqQLuvuJ

-to-

/ip4/104.131.131.82/tcp/4001/**p2p**/QmaCpDMGvV2BGHeYERUEnRQAwe3N8SzbUtfsmvsqQLuvuJ

Gateway: Subdomain Gateway



Starlog

Available as RSS

This is the IPFS Starlog, a series of communications about the [IPFS Project](#). These posts are written by the IPFS team, and members of the broader community. The subject matter is broad: project news, protocol explanations, stories, usage examples, application spotlights, and more.

Starlog entries: 176



Gateway: Directory Listing Refresh

The image shows two screenshots of the IPFS Gateway interface. On the left, a screenshot of a browser window displays a directory listing for a specific IPFS path. The title bar says "IPFS". The page header reads "Index of /ipfs/bafybeia6po64b6tfqq73lckadrhpihg2oubaxgqaoushquhcek46y3zumm". The list includes files like "about", "contact", "help", "ping", "quick-start", "readme", and "security-notes" with their respective file sizes. A large grey arrow points from this screenshot to the right one. On the right, another screenshot of the same browser window shows the same directory listing, but the file sizes have changed. For example, "about" is now 1.7 kB, "contact" is 189 B, "help" is 311 B, "ping" is 4 B, "quick-start" is 1.7 kB, "readme" is 1.1 kB, and "security-notes" is 1.2 kB. The IPFS logo is visible in the top right corner of the browser window.

File	Original Size	Refreshed Size
about	1.7 kB	1.7 kB
contact	189 B	189 B
help	311 B	311 B
ping	4 B	4 B
quick-start	1.7 kB	1.7 kB
readme	1.1 kB	1.1 kB
security-notes	1.2 kB	1.2 kB

IPNS: Transparent ENS



/ipns/ipfs.eth

IPNS:

Improved IPNS over PubSub

- Fetches the latest record from pubsub peers instead of the DHT.
- Ensures we remain connected to pubsub peers interested in the same IPNS key using the DHT.
- Will be stabilized in go-ipfs 0.6.0 (along with pubsub).

And more...

- Reduced badger memory usage.
- Flatfs improvements: avoid leaving temporary files behind and better handle file descriptor limits.
- Dial backoff logic is now per address instead of per-peer.
- Minimum RSA key size has been bumped to 2048.
- The default PubSub router has been changed from floodsub to gossipsub.
- API now only accepts POST requests.
- "Error: api not running" error has been killed off along with the `ipfs repo fsck` command.
- The `ipfs add` command has gained `--ignore` and `--ignore-rules-path` flags to ignore files on add.
- Significantly improved systemd support.
 - With systemd notification & socket activation support.
- The API can now listen on unix domain sockets.
- Docker container is built with OpenSSL by default.

And even more...

- Plugins (still suck)
 - MacOS support
 - New permanently unstable "internal" plugin type for ultra-privileged plugins with access to all internals.
 - Plugin configuration support.
- IPFS can be initialized with an existing config: `ipfs init /path/to/existing/config`
- Pins can be streamed with `ipfs pin ls --stream`
- Two repo migrations
 - Bootstrapper migration to migrate to the new bootstrappers. The old ones used 1024bit RSA keys.
 - Keystore migration to encode key names on disk for better cross-platform compatibility.
- Windows improvements
 - Additional antivirus workarounds in flatfs.
 - Double-clicking `ipfs.exe` starts the daemon.
 - `ipfs add -r` now properly recognizes hidden files on windows.
- And a *ton* of bug fixes.

Areas of Improvement



AutoNAT



Routing
Tables



Lookup
Algorithm

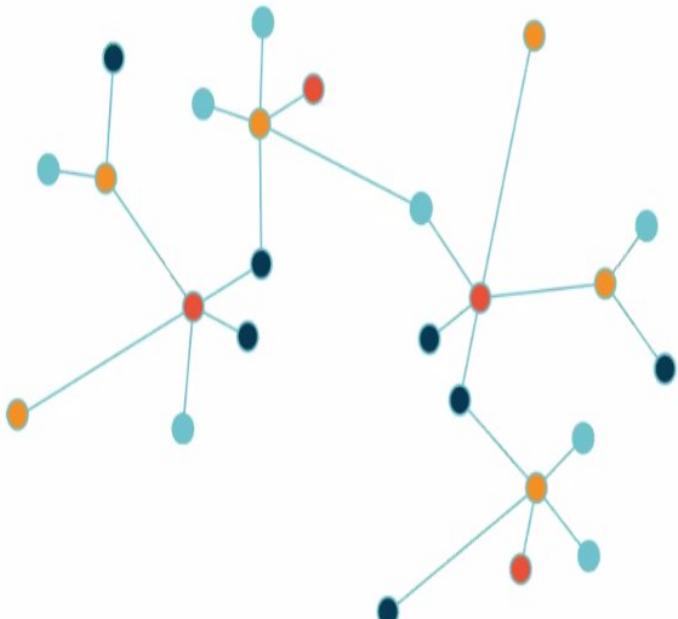


Testing

AutoNAT: Cleaning up the DHT



- IPFS's Kademlia DHT is a single network of peers where peers ask questions like: "Do you have some data, **X**, or know anyone who has it"?
- Problem:** What happens when not all peers can connect to all other peers? The network structure starts to degrade



AutoNAT:

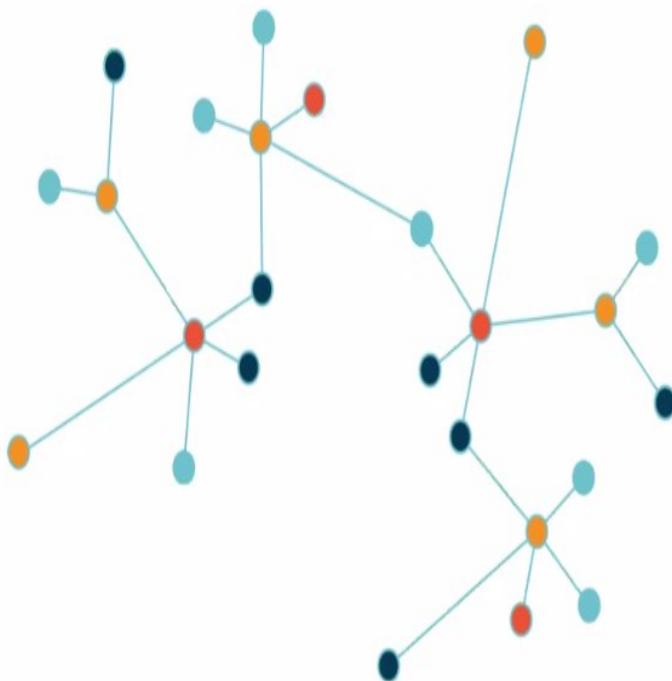
Cleaning up the DHT



- **Solution:**

- Ask other peers about our dialable status
- If we determine we're not dialable, stop advertising that we can respond to queries

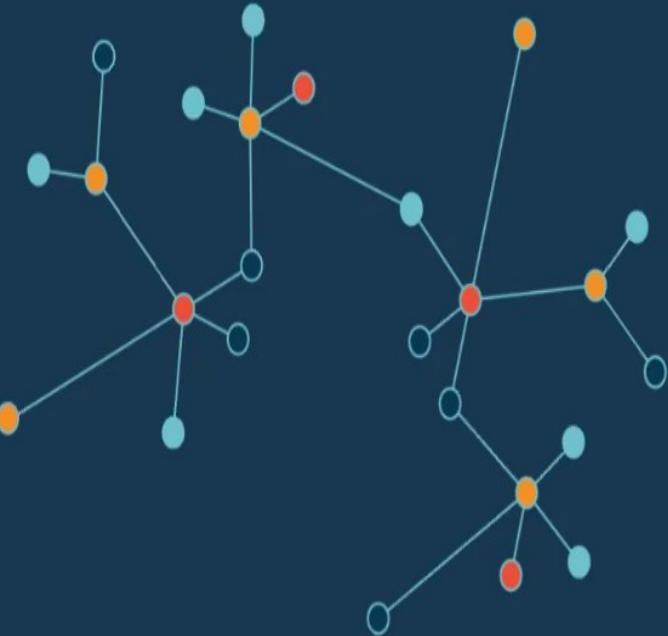
- Backup Plan: All peers should defensively examine their connections from other peers to determine if they look like they are connectable



Routing Tables: Keeping good Peers



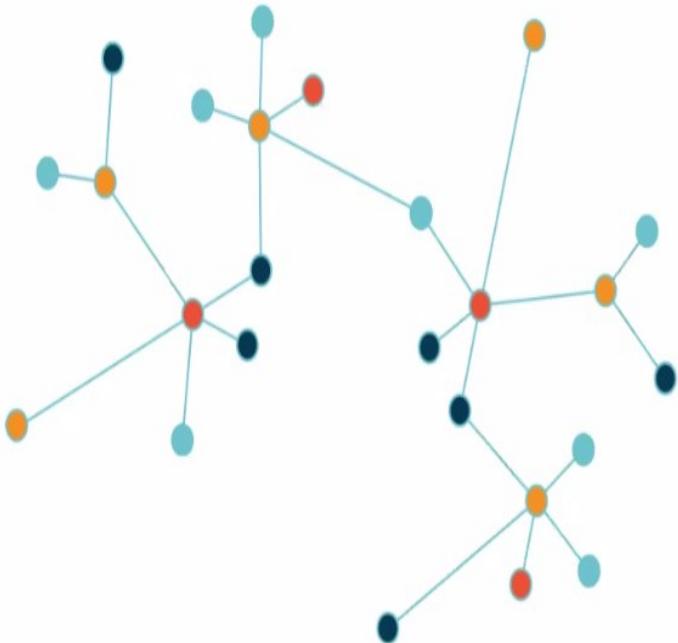
- Routing tables form the basis of how peers find each other in the DHT
- So many peers were **undialable** that we decided to only tell people about peers we had active connections to
- **Problem:** Forgetting Good Peers
 - Intermittent network disconnects
 - Heavy load of non-DHT connections



Lookup Algorithm: So, who do we ask?



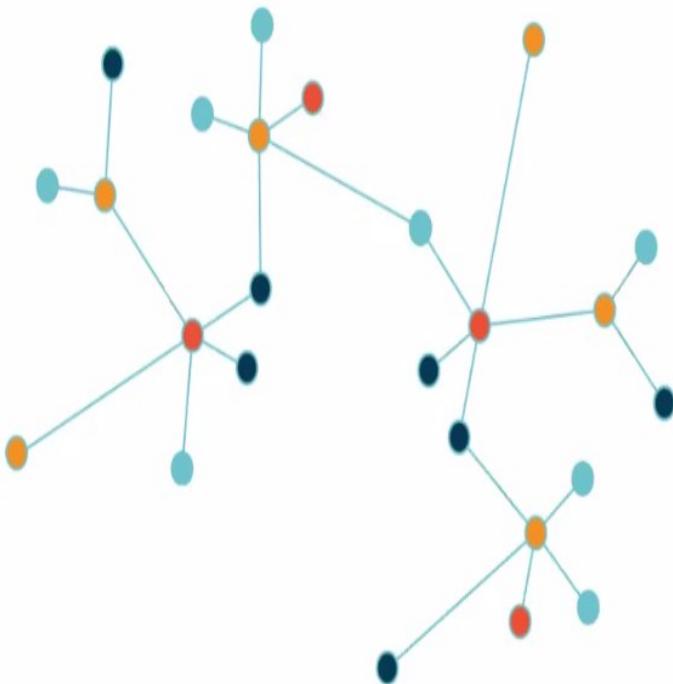
- The Kademlia lookup logic repeatedly asks peers: “Do you know anyone closer to you than **X**? ”
- **Problem:**
 - Routing tables full of “bad” peers cause the query to take a very LONG time.
 - Many failed dials to peers
 - Decreases confidence that we have reached the closest peers to **X**
 - Requires searching more of the network



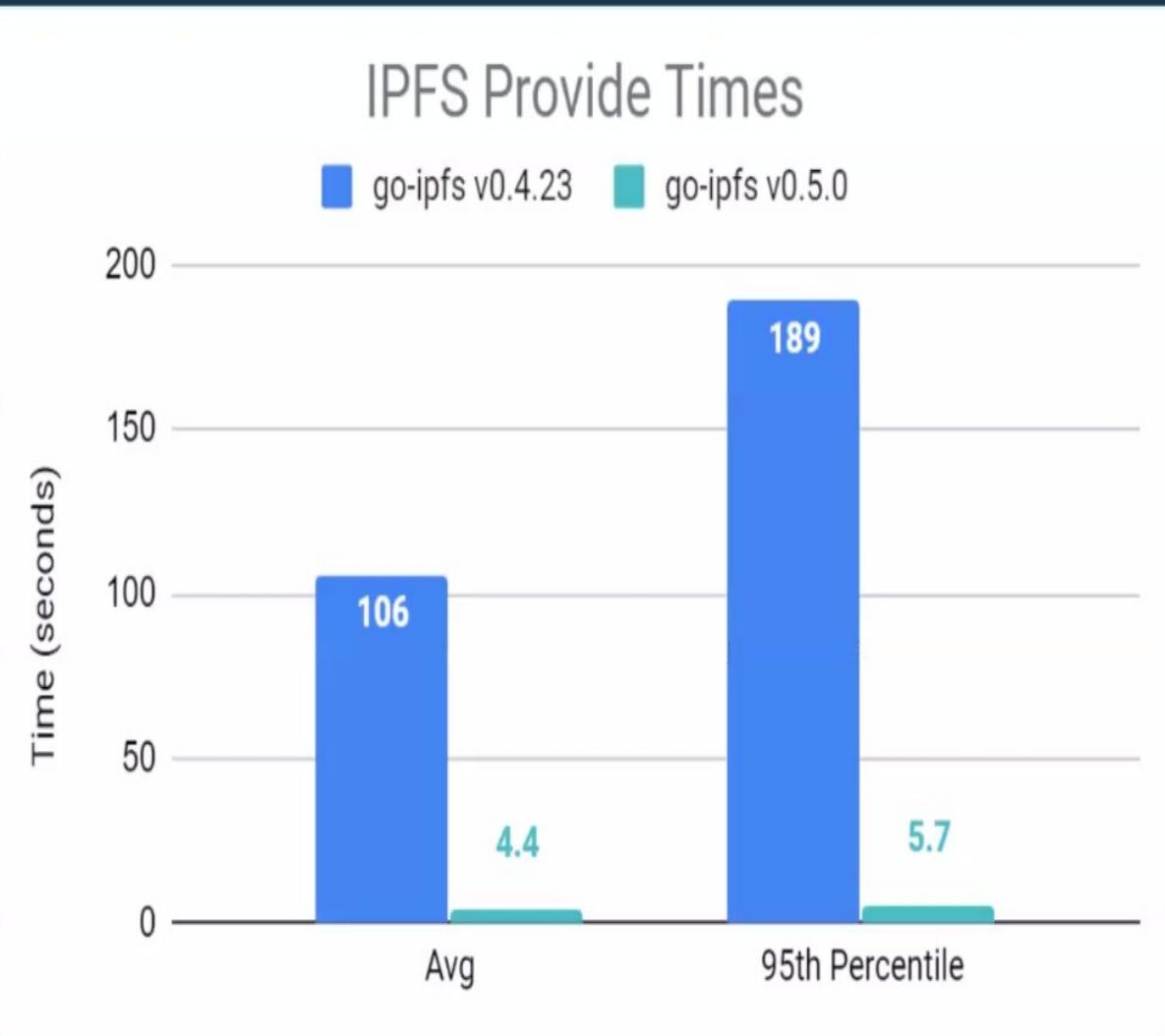
Lookup Algorithm: So, who do we ask?



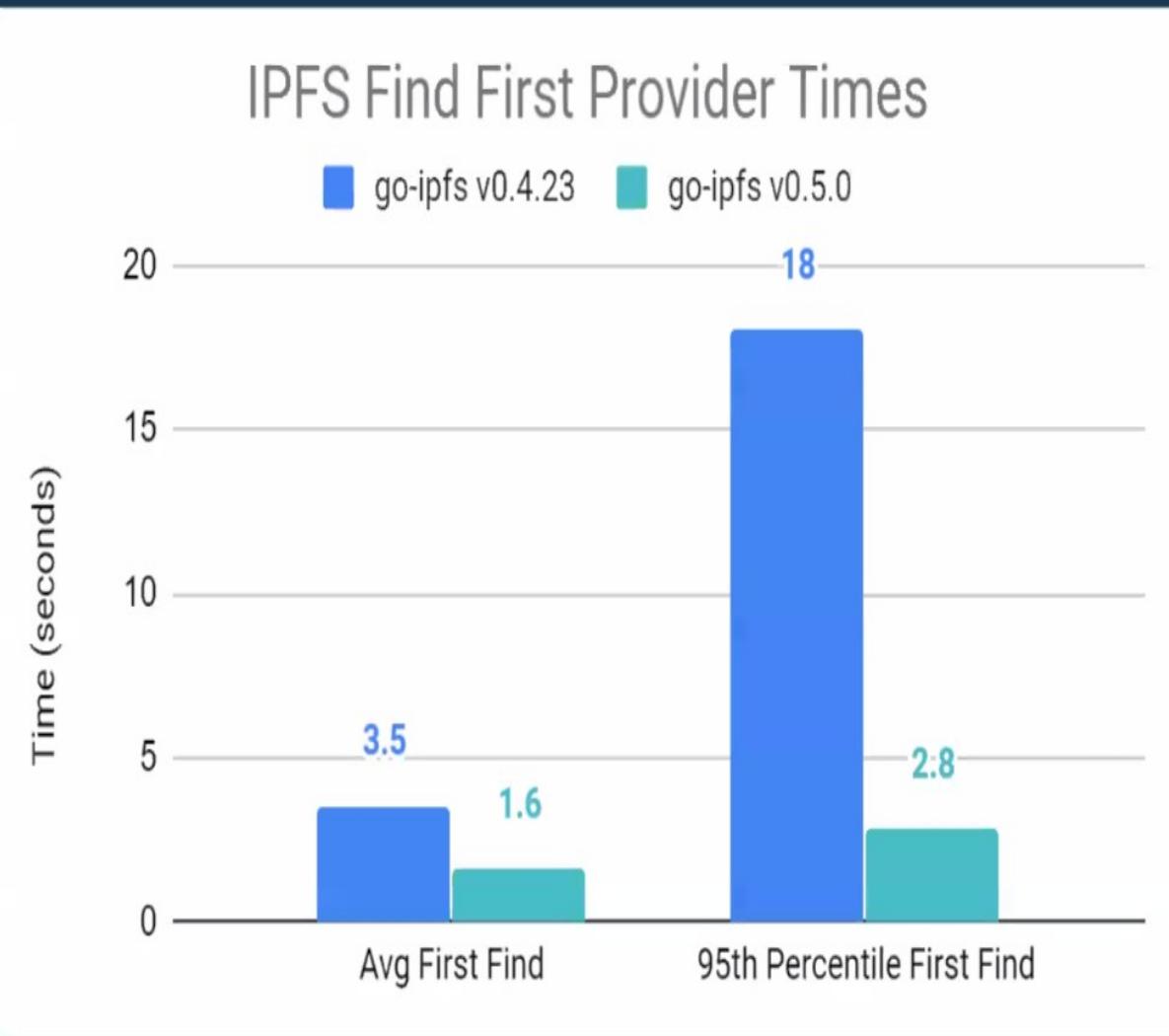
- **Solution:** With improved routing tables we can...
 - Query the closest peers we know to **X**, until they don't know anyone closer



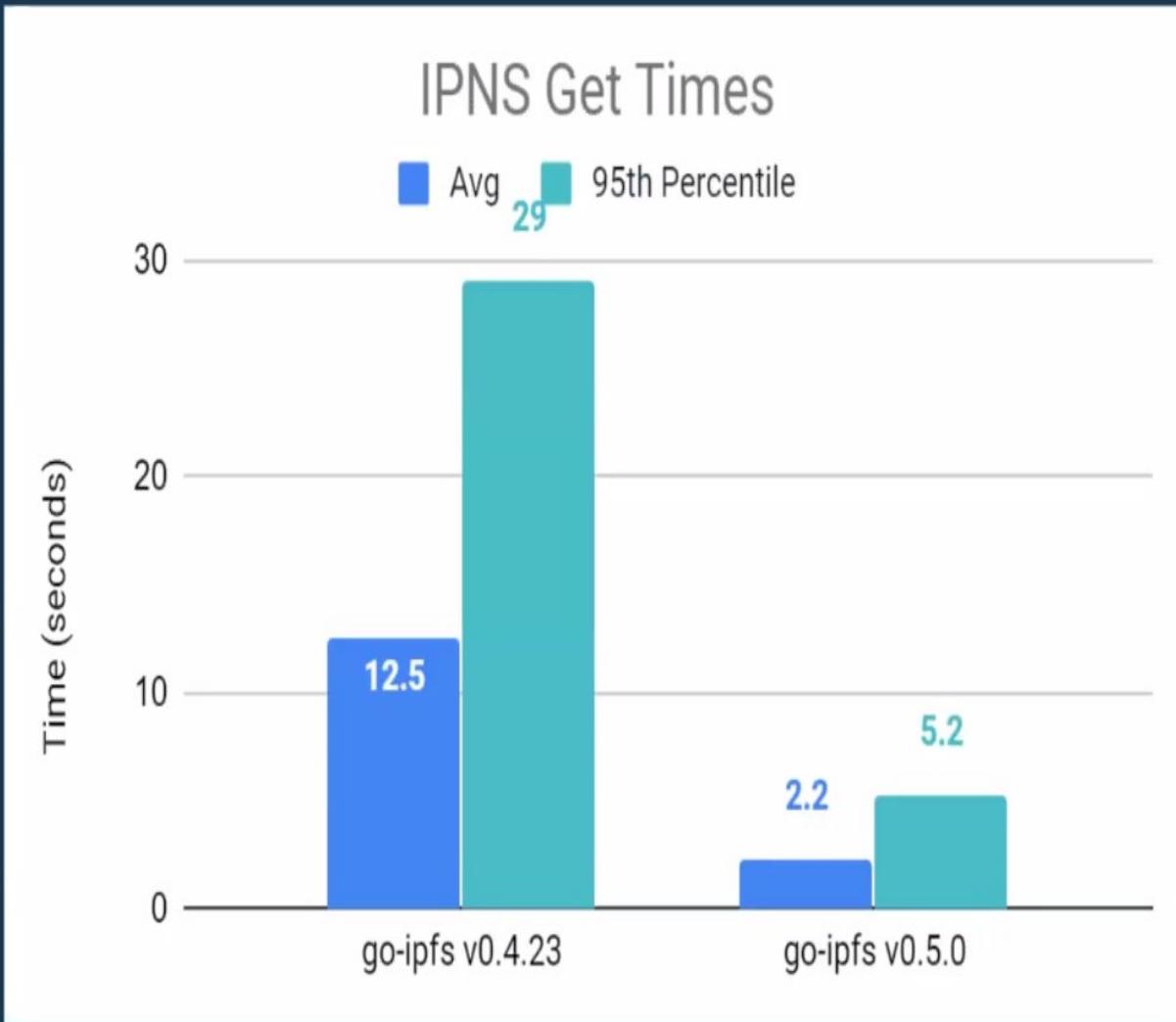
Testing: DHT Results



Testing: DHT Results



Testing: DHT Results



Everyone	Well done Adin!	
Zachary Robinson (Co-host)	From Juan Benet to Everyone :	08:53 WB
Molly Mackinlay (Co-host)	It's awesome to see so many people around	
Jacob Heun (Co-host)	From mike to Everyone :	08:53 WB
Andrew Nesbitt (Co-host)	hella down for IPFS swag when IPFS online shop?	
Adin Schmahmann (Co-ho...)	From Mk to Everyone :	08:53 WB
Will Scott (Co-host)	not sure whether it's right time to ask, could someone describe in more detail what happens when a node containing data leaves	
Stebalien (Co-host)	From Dadepo Aderemi to Everyone :	08:53 WB
Alan Shaw (Co-host)	How are these improvements in the go implementation going to be reflected in the ongoing Rust effort? Any conscious plans/thoughts about this?	
Marcin Rataj (lidel) (Co-ho...)	From Juan Benet to Everyone :	08:54 WB
Matt Ober (Co-host)	^ I think they'll build the latest versions of algorithms in rust, so they should be reflected from the get-go	
Carson Farmer (Co-host)	From Stebalien to Everyone :	08:54 WB
IPFS 0.5 Launch Meetup (...	Mk: If nobody else has the data, it'll be gone from the network.	
	From Juan Benet to Everyone :	08:54 WB
	awesome, great work	
	From Darren Tapp to Everyone :	08:55 WB
	How are conflicts in the name service handled? Two claims on the same name.	
	From Mark Henderson to Everyone :	08:56 WB
	It's true! Trying to stay abreast. thank you molly	
	From Petar Dochev to Everyone :	08:56 WB
	does the DHT still use Kademlia?	
	From Diaa Kasem to Everyone :	08:56 WB
	Any plans to integrate with OS layer directly .. instead of having to install IPFS ?	
	From Dadepo Aderemi to Everyone :	08:57 WB
	Thanks for the answer :)	
	From Adin Schmahmann to Everyone :	08:57 WB
	Darren: You can't really have two claims on the same name. Either you are using DNSLink where you have the same guarantees as DNS, or you are using IPNS where the name is controlled by access to a private key.	
	From Stebalien to Everyone :	08:57 WB
	Darren Tapp: The names are cryptographic hashes of public keys so there are no clashes.	
To: Everyone		More ▾



ThreadsDB



A secure, decentralized, p2p database built on IPFS, Libp2p, and IPLD

- Easy MongoDB-like developer interface.
- Modular ACL, Identity, CRDTs and more.
- Encryption design that enables private sharing **and** 3rd party pinning, relay, archival, etc.

ThreadsDB



```
const db = new Database(...)  
const schema = { ...json-schema.org }  
const Astronauts = await db.newCollection('Astronauts', schema)  
db.on('Astronauts.**', update => {  
  console.log(update) // subscribe to updates  
})  
await Astronauts.insert(  
  { ID: '', missions: 2, name: 'Buzz Aldrin' },  
  { ID: '', missions: 5, name: 'Christina Koch' },  
  { ID: '', missions: 2, name: 'Sally Ride' },  
  { ID: '', missions: 7, name: 'Jerry Ross' },  
)  
const all = Astronauts.find(  
  { $or: [{ missions: { $gt: 2 } }, { name: 'Sally Ride' }] },  
  { sort: { missions: -1 } }  
)  
for await (const { key, value } of all) {  
  console.log(value)  
}
```



ThreadsDB



links:

Whitepaper: docs.send.com/view/gu3ywqi

Go impl: github.com/textileio/go-threads

JS impl: github.com/textileio/js-threads

*Hey, but what about files and
directories?*

Buckets



Dynamic folders over ThreadsDB

- Collaborative, multi-user folders.
- Automatically persisted on remote IPFS peers.
- Interop FTW! Every folder is simultaneously (a) an instance in your Thread; (b) an HTTP available domain; (c) an IPNS address on the IPFS network!!!

The Hub



Remote IPFS and ThreadsDB peers that will radically change the way you can build apps.

- Developers can access free remote services to persist Buckets and encrypted Thread data.
- Connect your Threads with one line to start storing and relaying data for collaborators. Scale!
- Developers generate App Keys that allow their users to access those same resources!

The Hub



Remote IPFS and ThreadsDB peers that will radically change the way you can build apps.

- Developers can access free remote services to persist Buckets and encrypted Thread data.
- Connect your Threads with one line to start storing and relaying data for collaborators. Scale!
- Developers generate App Keys that allow their users to access those same resources!

The Hub

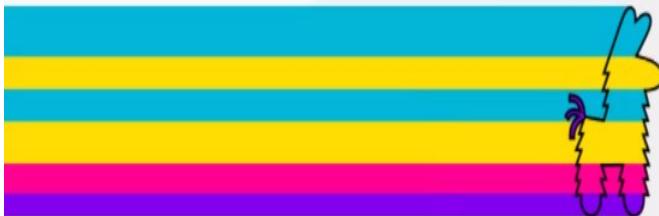


User-controlled data, paid for with
Filecoin...

- App Keys provide end-user ownership of their data, work in browser and mobile apps, and are interoperable with any pk or email user identities.
- Powergate allows Buckets and Threads to start running on Filecoin at launch of Mainnet.

Trust the data

- Content Identifiers (CIDs)
- Trust the data without trusting where it comes from



Pinata and IPFS 0.5 - DHT Improvements

- Faster Content Announcing
- Faster Content Discovery



Pinata and IPFS 0.5 - Bitswap Improvements

- Faster Content Delivery
- More Efficient Content Delivery
 - Less duplicate blocks being sent



```
ng> ipfs --version
ipfs version 0.5.0
> ipfs init
initializing IPFS node at /home/lidel/.ipfs
generating 2048-bit RSA keypair...done
peer identity: QmcbijQnfz7xeZj7qW6EyVxnE19i1sG8ua7FWZWKBqafst
to get started, enter:
```

```
ipfs cat /ipfs/QmQPeNsJPyVWPFDVHb77w8G42Fvo15z4bG2X8D2GhfbSXc/readme
```

```
> ipfs config --json Gateway.PublicGateways '{
  "example.com": {
    "UseSubdomains": true,
    "Paths": ["/ipfs", "/ipns"]
  }
}'
> vim ~/.ipfs
```

```
"API": {  
    "HTTPHeaders": {}  
},  
"Addresses": {  
    "API": "/ip4/127.0.0.1/tcp/5001",  
    "Announce": [],  
    "Gateway": "/ip4/127.0.0.1/tcp/8080",  
    "NoAnnounce": [],  
    "Swarm": [  
        "/ip4/0.0.0.0/tcp/4001",  
        "/ip6::/tcp/4001"  
    ]  
},  
"AutoNAT": {},  
"Bootstrap": [  
    "/ip4/104.131.131.82/tcp/4001/p2p/QmaCpDMGvV2BGHeYERUEnRQAwe3N8SzbtfsmsqQLuvuJ",  
    "/dnsaddr/bootstrap.libp2p.io/p2p/QmNnooDu7bfjPFoTZxMNLWUQJyrVwtbZg5gBMjTezGAJN",  
    "/dnsaddr/bootstrap.libp2p.io/p2p/QmQCU2EcMqAqQPR2i9bChDtGNJchTbq5TbXJJ16u19uLTa",  
    "/dnsaddr/bootstrap.libp2p.io/p2p/QmbLHAnMoJPWSCR5Zhtx6BHJX9KiKNN6tpvbUcqanj75Nb",  
    "/dnsaddr/bootstrap.libp2p.io/p2p/QmcZf59bWwK5XFi76CZX8cbJ4BhTzzA3gU1ZjYZcYW3dwt"  
],  
"Datastore": {
```

```
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/172.17.0.1/tcp/4001
Swarm announcing /ip4/192.168.1.17/tcp/4001
Swarm announcing /ip4/79.191.39.212/tcp/5145
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

```
> chromium --user-data-dir=$(mktemp -d) --proxy-server="http://127.0.0.1:8080" "http://example.com/ipfs/bafybeie  
mxf5abjwjbjkoz4mc3a3dla6ual3jsgpdr4cjr3oz3evfyavhwq/wiki/Vincent\_van\_Gogh.html"
```

```
bck-i-search: chrom_
```

```
Swarm listening on /ip6::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/172.17.0.1/tcp/4001
Swarm announcing /ip4/192.168.1.17/tcp/4001
Swarm announcing /ip4/79.191.39.212/tcp/5145
Swarm announcing /ip6::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

```
> chromium --user-data-dir=$(mktemp -d) --proxy-server="http://127.0.0.1:8080" "http://example.com/ipfs/bafybeie\_mxf5abjwjbikoz4mc3a3dla6ual3jsgpdr4cjr3oz3evfyavhwq/wiki/Vincent\_van\_Gogh.html"
```

```
[27789:27789:0501/204338.118497:ERROR:sandbox_linux.cc(374)] InitializeSandbox() called with multiple threads in process gpu-process.
```

```
[27793:27833:0501/204338.125449:ERROR:ssl_client_socket_impl.cc(941)] handshake failed; returned -1, SSL error code 1, net_error -107
```

```
[27793:27833:0501/204338.242672:ERROR:ssl_client_socket_impl.cc(941)] handshake failed; returned -1, SSL error code 1, net_error -107
```

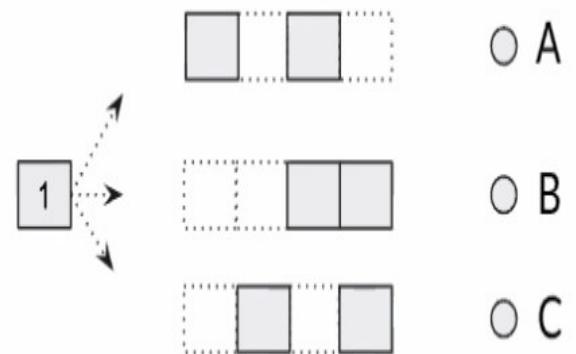
```
[27793:27833:0501/204339.141595:ERROR:ssl_client_socket_impl.cc(941)] handshake failed; returned -1, SSL error code 1, net_error -107
```



What's New

- Less duplicate blocks
 - Find out who has blocks
 - Only request the blocks from the peers who have them

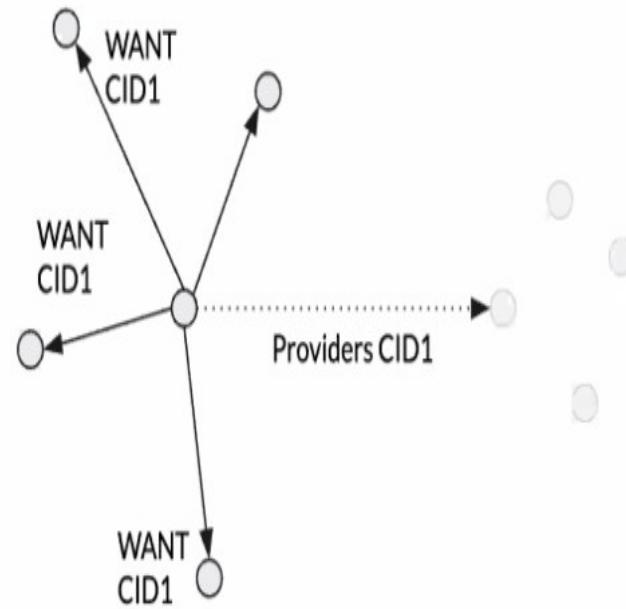
- Transfer files faster
 - Less aggressive rate limiting
 - Find out if a peer doesn't have a block (so we can ask someone else / broadcast)



Bitswap - Discovery

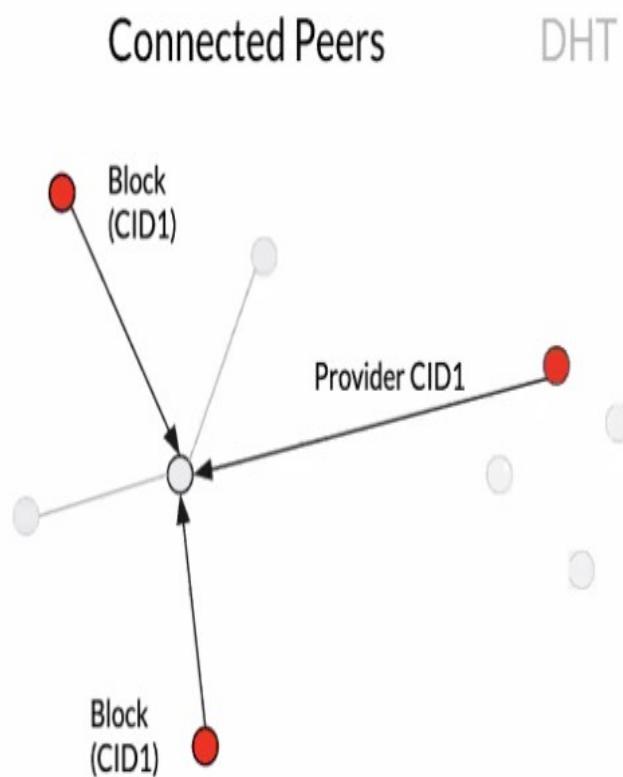
- Broadcast WANT to connected Peers
- If there's no response, ask DHT who has root CID

Connected Peers DHT



Discovery - Sessions

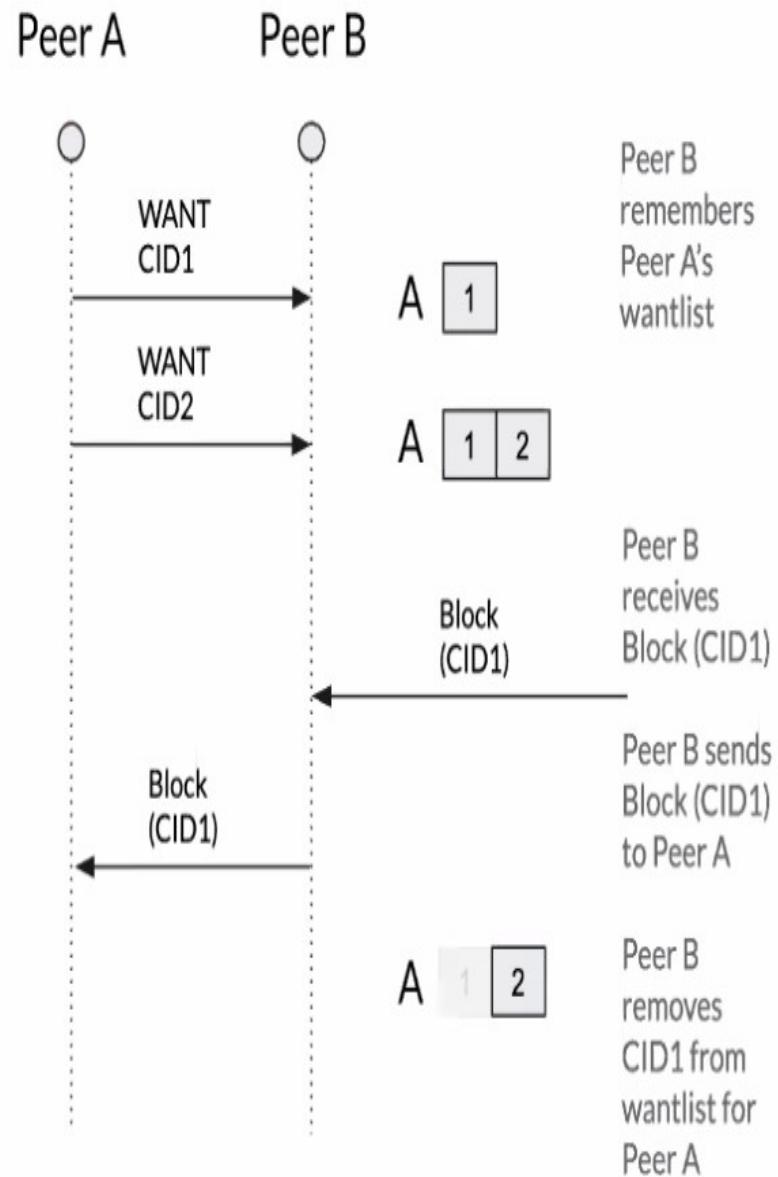
- Peers who respond are added to the Session
- Subsequent requests are sent only to peers in the session





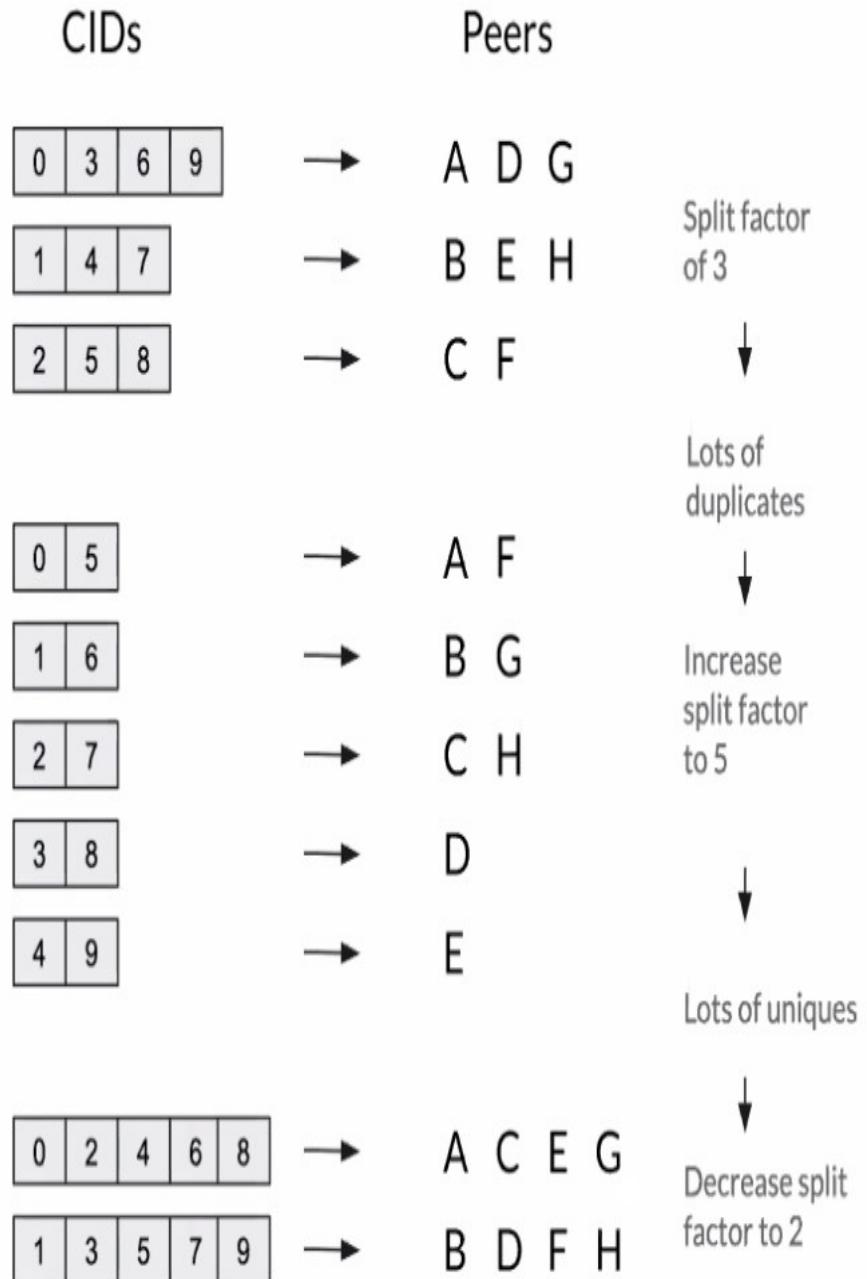
Wants

- Nodes send WANT messages to peers
- Each node remembers the want list for each of its peers
- The wantlist is discarded when the peer disconnects



Request Splitting (Old Bitswap)

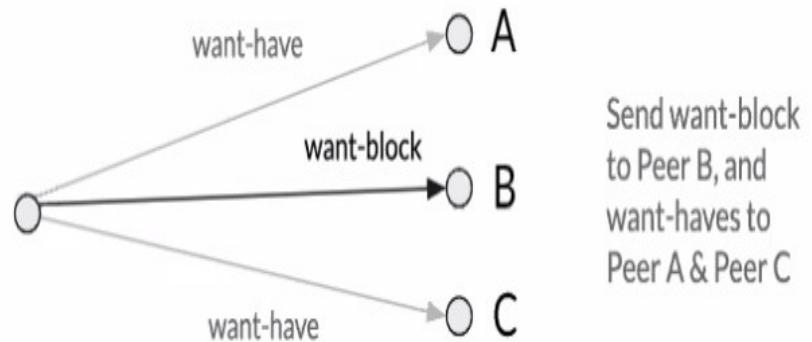
- Peers in the session are ordered by latency
- Request is split across peers
- Split factor varies according to uniq/dup ratio



Protocol Extensions

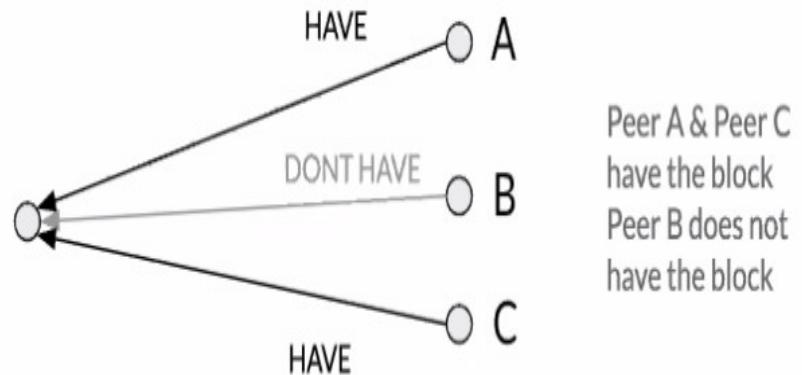
want-haves

- Send want-haves to all peers
- Send an optimistic want-block to one peer



HAVE / DONT HAVE

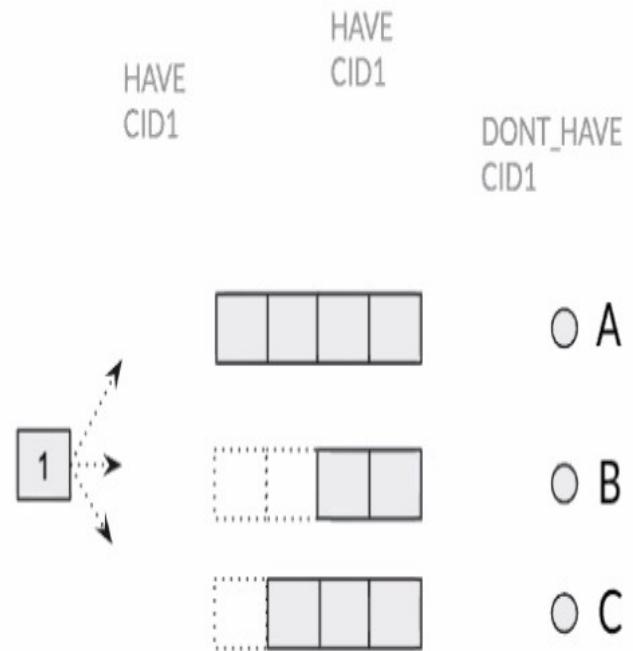
- Peers respond with HAVE or DONT HAVE
- Send want-block peer that has block



Implementation Improvements

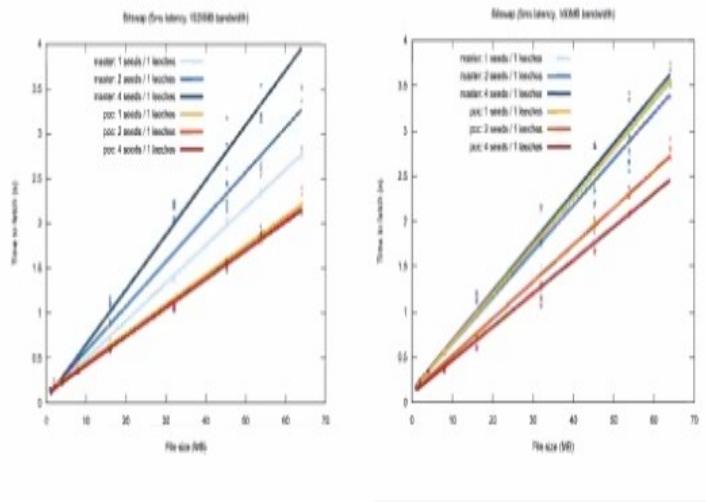
Peer Selection

- When choosing which peer to send an optimistic want-block to:
- Prioritize peers that
 - Sent a HAVE for the CID
 - Is a Provider for the CID
- Ignore peers that sent a DONT_HAVE for the CID
- If no peer has sent a HAVE for the CID, select peer probabilistically according to which peers sent us the most blocks for other wants in the session

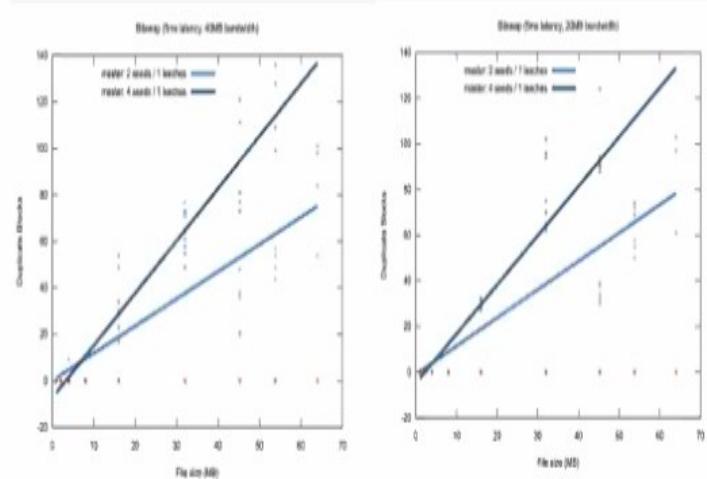
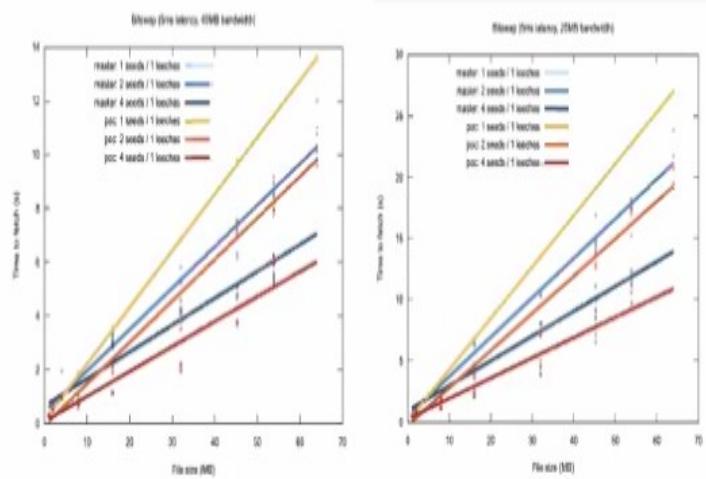
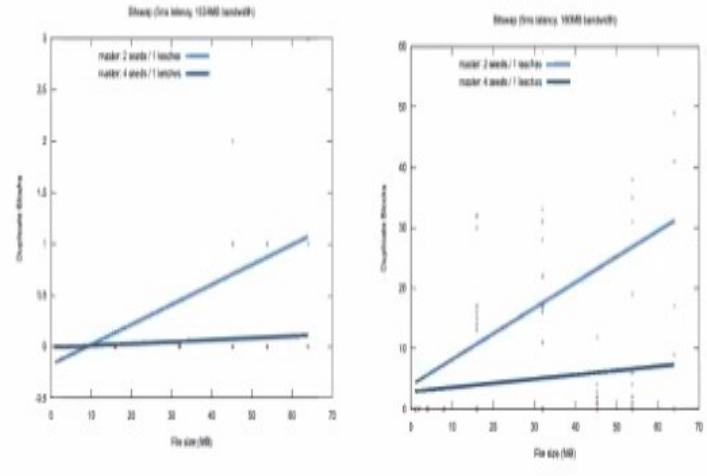


Results

5ms latency, variable bandwidth: 1 seed, 2 seeds, 4 seeds



Duplicate blocks + 5ms latency, variable bandwidth: 1 seed, 2 seeds, 4 seeds





Logout

Control Panel

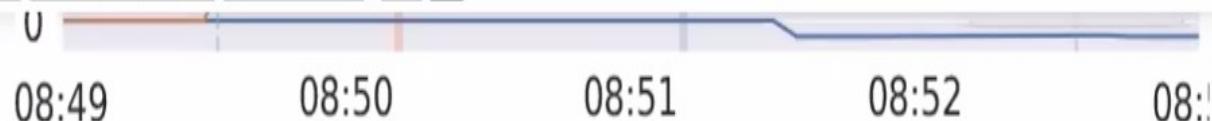
File Edit View Insert Cell Kernel Widgets Help

Not Trusted

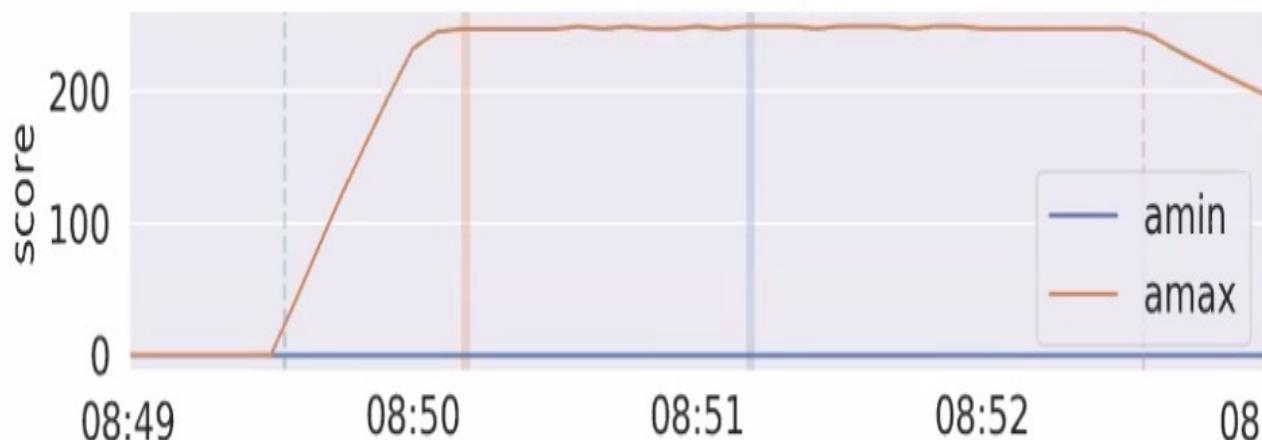
Python 3



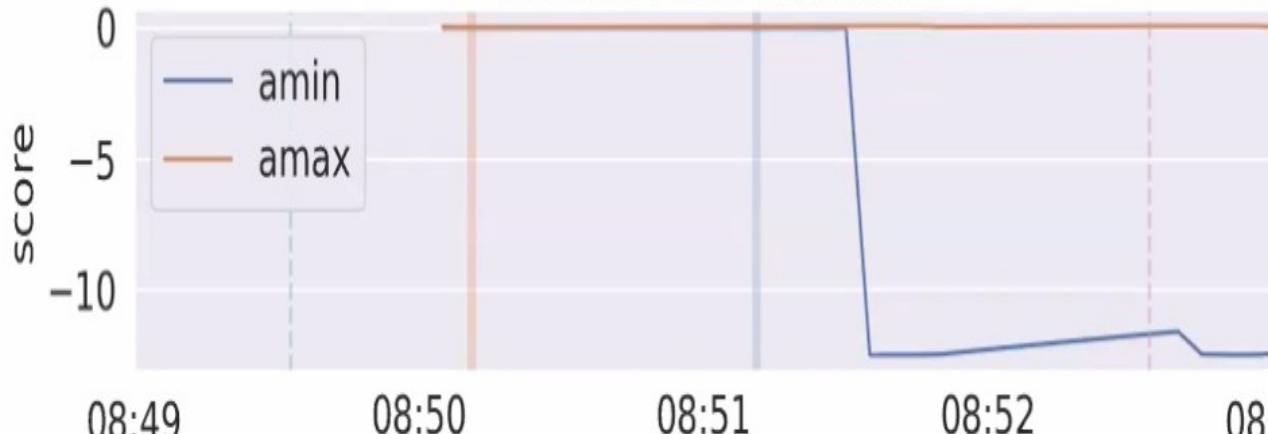
Memory: 24614 MB



Honest Peers min/max



Attacker Peers min/max



warmup complete

File Edit View Insert Cell Kernel Widgets Help

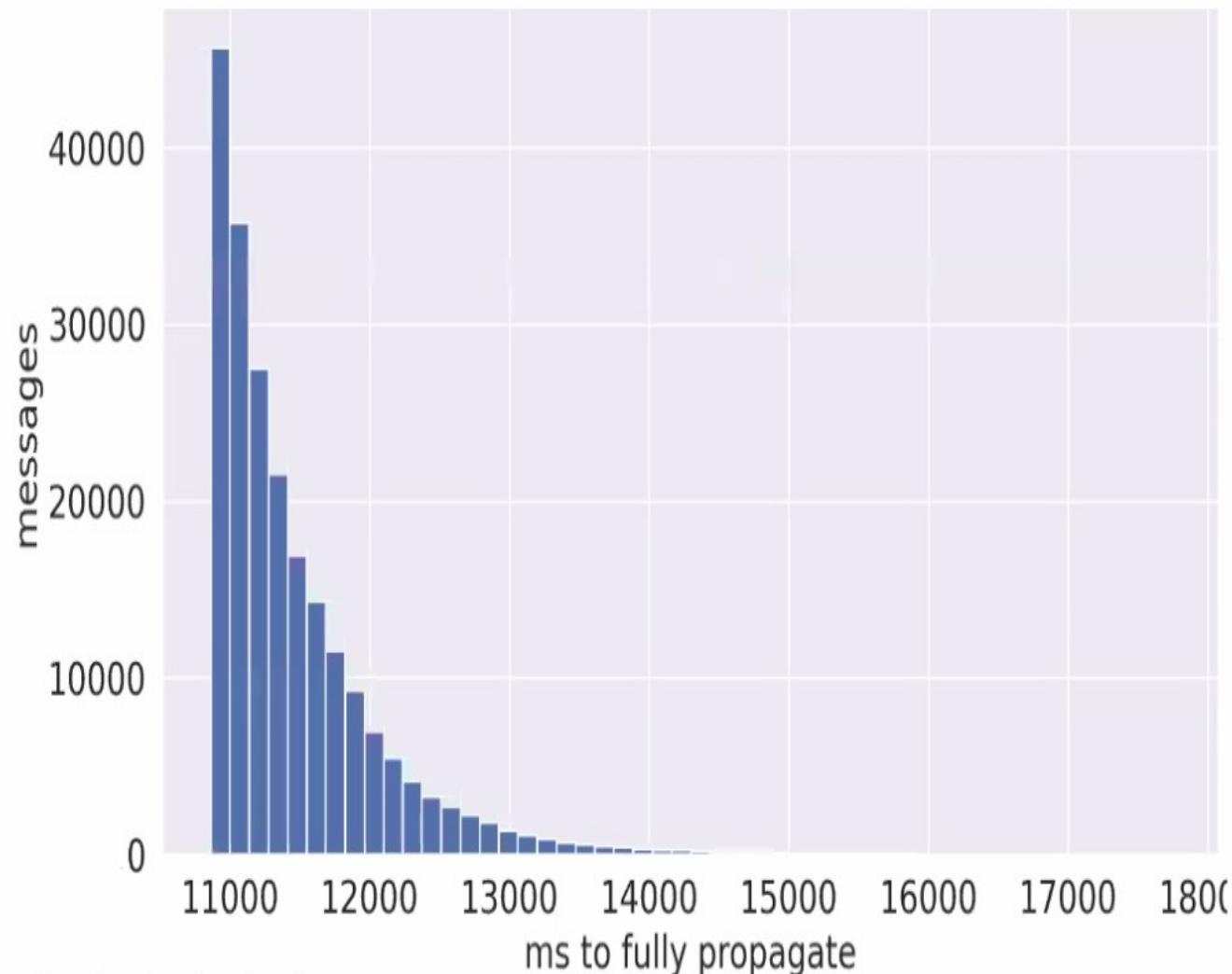
Not Trusted

Python 3



Memory: 24582 MB

Latency PDF above p99 (10861.00ms)



JAMStack



3

- JAM stands for JavaScript, APIs, and Markup (<https://jamstack.org/>)
- Why does it matter for IPFS?
- We use Gatsby

Description, note or comment (not more than 1 row)

stake.fish is powered by IPFS completely



4

- stake.fish gets statically compiled at every commit with a full CI/CD pipeline
- stake.fish site gets pinned on pinata.cloud
- _dnslink TXT record gets updated
- Our content is served directly from a IPFS gateway

Description, note or comment (not more than 1 row)



If you would like to learn more about stake.fish and staking in general, please don't hesitate to reach out.

We would like to contribute to the Wikipedia on IPFS project. Any developers interested?

We are hiring!

An icon of a speech bubble containing three dots, representing email or communication.

hi@stake.fish
jobs@stake.fish