

# Tool Spec: DHT Tracer

A tool to trace & debug the libp2p/ipfs main DHT.

1. [Project Management](#)
2. [Urgency](#)
3. [Architecture](#)
4. [Backend](#)
5. [Frontend](#)
6. [Data Formats](#)

## Project Management

- Owner: @jbenet - connect w/ him if you are going to work on this.
- Repos
  - <https://github.com/libp2p/dht-tracer1>
  - <https://github.com/cwaring/dht-trace>

## Urgency

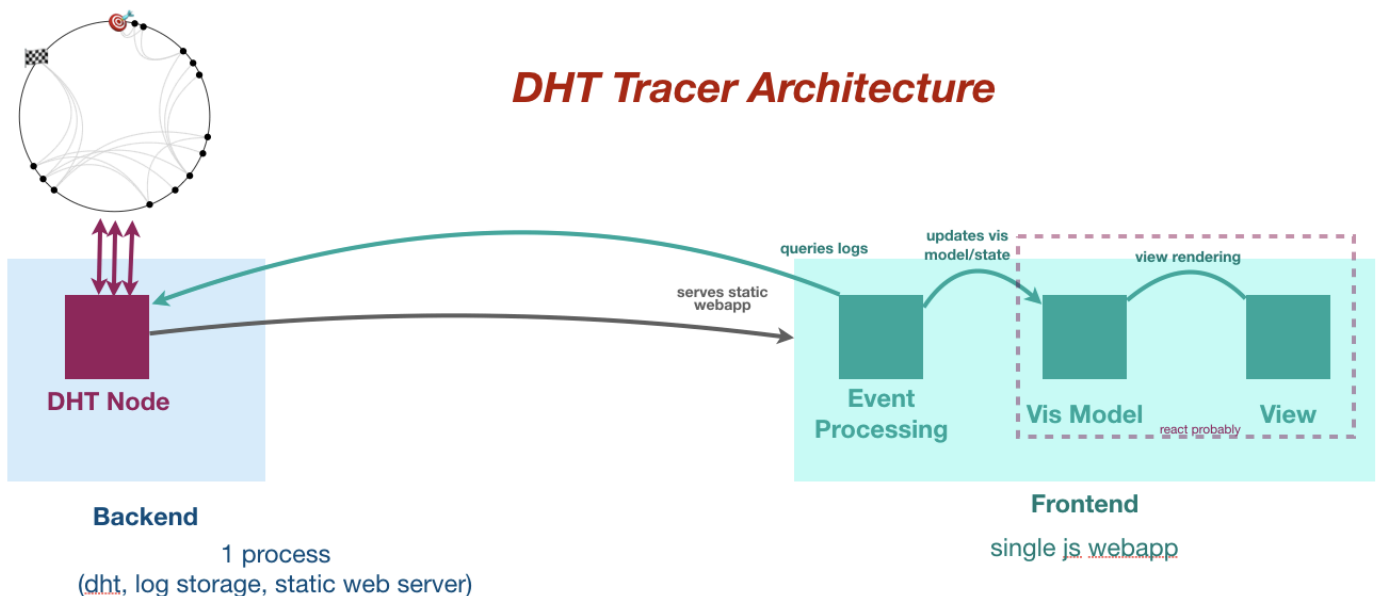
We have on-going service degradation (and somewhat of an outage) due to our DHT scaling 30x. Things are broken and we need to debug what's going on. Being able to visualize what's going on is a very useful way to figure out what the problems are. We believe making this tool should be accomplishable in 1 day, at least a rough mvp.

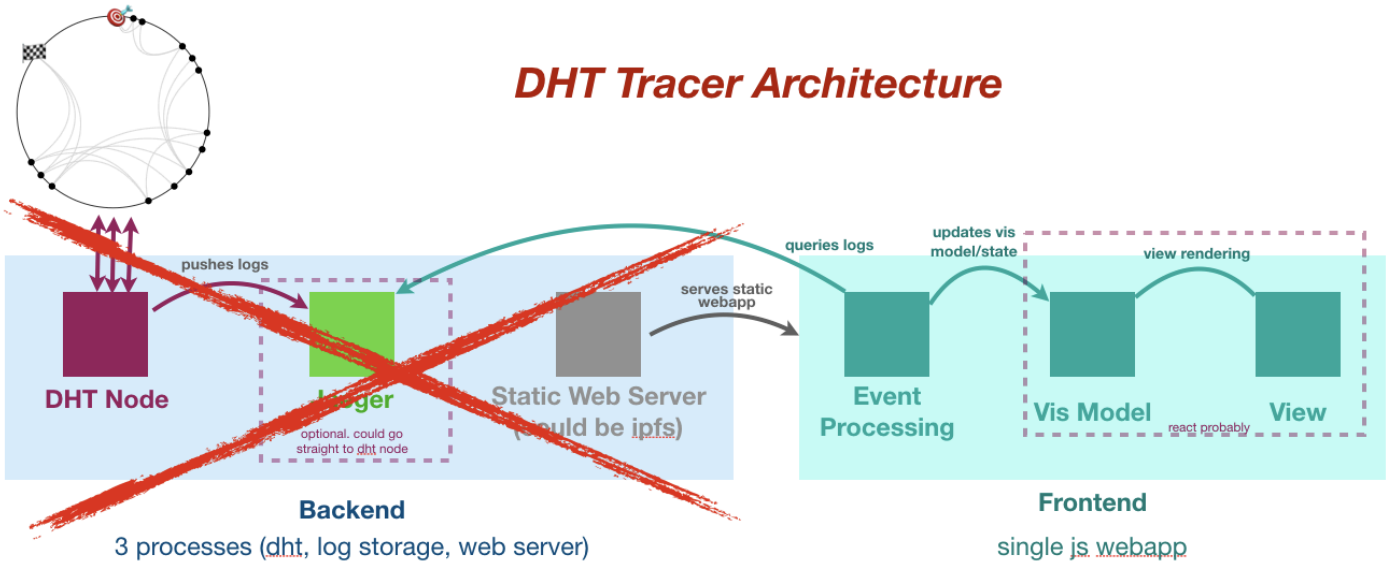
## Architecture

Two parts. See more below.

- **Backend:** Runs a DHT node (somehow, many kinds of ways), produces an event stream for the frontend.
- **Frontend:** Consumes event stream, transforms, and visualizes it. Used by the developer to debug what's going on.

We went for this in the end:



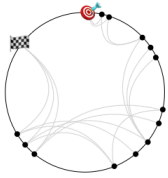


## Backend

- **DHT node, which outputs trace events**
  - A Go tool like dht-node, dht-simple-tests, or even go-ipfs
  - Outputs trace events (OpenCensus or OpenTrace)
  - <https://github.com/libp2p/dht-tracer1/>
- ~~(optional) Jaeger, to aggregate and store the traces~~
  - ~~We don't need this, but can make this easier~~

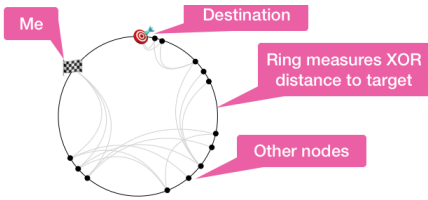
## Frontend

- **A WebApp that filters, transforms, and visualizes the events.**
- **Data processing**
  - Connects to the backend (either to the DHT node directly, or to Jaeger) to get event stream
  - Transforms event stream to something the visualization can consume.
- **Visualization**
  - A visualization similar to browser dev tools (network tab), with a gantt/waterfall like chart. (see below).
  - Visualization should be informative and easy for developers/debuggers to expand/augment by contributing to the code
  - See figures below.



XOR Hops

| ▼ Query Qmfoobar  | 0   | 15 | Seen: 100, Queried: 50, Dialed: 30, ToQuery: 10, Remaining: 15, Results: 3 | 7.5s, 3 rec, 258 peers    |
|-------------------|-----|----|--|---------------------------|
| Peer Qmfoo1       | 186 | 0  | 310ms, 2/5 peers   |                           |
| Peer Qmfoo2       | 163 | 0  | 325ms, 3/3 peers   |                           |
| Peer Qmfoo3       | 170 | 0  | 50ms, 10/10 peers  |                           |
| Peer Qmfoo4       | 102 | 1  |  | ✗ 3s                      |
| Peer Qmfoo5       | 95  | 1  | 52ms, 1 peer   |                           |
| Peer Qmfoo6       | 86  | 1  |  | ✗ 3s                      |
| Peer Qmfoo7       | 82  | 1  |  | ✗ 3s                      |
| Peer Qmfoo8       | 77  | 1  | 894ms, 8/8 peers   |                           |
| Peer Qmfoo7 ⚠ dup | 82  | 2  | 700ms, 0/0 peers   |                           |
| Peer Qmfoo10      | 40  | 2  | 700ms, 1/7 peers   |                           |
| ⋮                 |     |    |  |                           |
| Peer Qmfoo31      | 14  | 9  | 1.2s, 2/3 peers  |                           |
| Peer Qmfoo32      | 15  | 10 | 600ms, 1/8 peers   |                           |
| ⋮                 |     |    |  |                           |
| Peer Qmfoo82      | 0   | 15 |  | ✓ 894ms, 3 rec, 1/7 peers |



Success  
Runtime  
Records found, if any  
Peers queried

XOR Hops

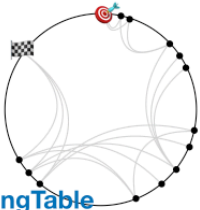
| ▼ Query Qmfoobar  | 0   | 15 | Seen: 100, Queried: 50, Dialed: 30, ToQuery: 10, Remaining: 15, Results: 3 | 7.5s, 3 rec, 258 peers    |
|-------------------|-----|----|--|---------------------------|
|                   |     |    | <# new peers returned> / <# of peers returned>                             |                           |
| Peer Qmfoo1       | 186 | 0  | 310ms, 2/5 peers   |                           |
| Peer Qmfoo2       | 163 | 0  | 325ms, 3/3 peers   |                           |
| Peer Qmfoo3       | 170 | 0  | 50ms, 10/10 peers  |                           |
| Peer Qmfoo4       | 102 | 1  |  | ✗ 3s                      |
| Peer Qmfoo5       | 95  | 1  | 52ms, 1 peer   |                           |
| Peer Qmfoo6       | 86  | 1  |  | ✗ 3s                      |
| Peer Qmfoo7       | 82  | 1  |  | ✗ 3s                      |
| Peer Qmfoo8       | 77  | 1  | 894ms, 8/8 peers   |                           |
| Peer Qmfoo7 ⚠ dup | 82  | 2  | 700ms, 0/0 peers   |                           |
| Peer Qmfoo10      | 40  | 2  | 700ms, 1/7 peers   |                           |
| ⋮                 |     |    |  |                           |
| Peer Qmfoo31      | 14  | 9  | 1.2s, 2/3 peers  |                           |
| Peer Qmfoo32      | 15  | 10 | 600ms, 1/8 peers   |                           |
| ⋮                 |     |    |  |                           |
| Peer Qmfoo82      | 0   | 15 |  | ✓ 894ms, 3 rec, 1/7 peers |

Duplicate peer queried (error)

Started Dialing Connected dht req Req end, returned peers

Has records

XOR distance to target # of hops in this query



## Potential react components

TracingTable



## Data Formats

We use two data formats, one that's coming out of the dht node (hard to change), and one that's going into the visualization (easy to change, but should be tuned for easy/fast visualization processing). We can transform the data in the middle.

- **DHT Node output event stream**
  - Trace events from OpenCensus (or OpenTrace?)
  - See [data types](#)
- **Visualization input state**
  - TBD (as needed by vis)
  - See [data types](#) and [example data](#)