Reprovide Sweep

Opening the DHT to large content providers

Gui Michel
@guissou

Probelab,
Protocol Labs

IPFS ÞING

Protocol Labs

IPFS Thing
16th April 2023

DHT Provide Process

When a node wants to advertise to the DHT that it provides `CID`

- DHT lookup to find the `repl` (20) closest peers
  - Number of hops: ~ 3
  - Number of inflight messages: ~ 10
  - Total number of sent messages: ~ 30

  tes
- Allocate the Provider Record to these `repl` closest peers

DHT Provide Process

- Number of connections opened per provide: ~ 35
  - Assume that we already have an open connection for peers in the first hop
  - Intermediary lookup peers: ~ 15
  - `repl` closest peers: 20
- Number of messages sent per provide: ~ 70
  - DHT Lookup: $\sim 30 + 20$
  - Provide messages: 20

Large Content Providers

Large content provider reproviding 1 Billion CIDs every `ReprovideInterval` (22h)

- Number of opened connections: $\sim 35B \approx 450'000/s$
- Number of messages sent: $\sim 70B \approx 900'000/s$

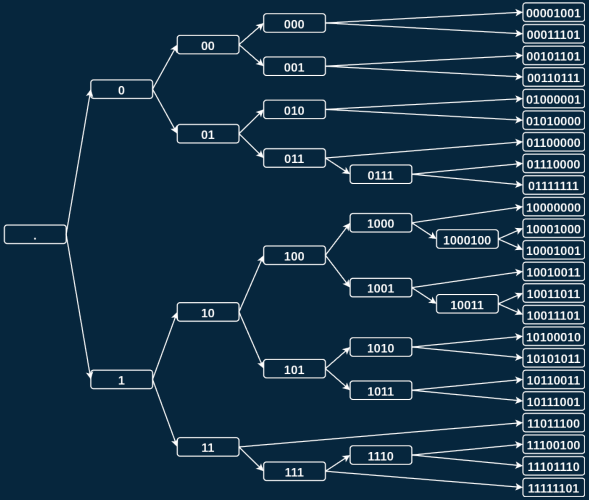
Optimizing the `Reprovide` operation

- More CIDs to reprovide than DHT Servers
- $\Rightarrow$ multiple CIDs are allocated on the same DHT Server
- Group CIDs allocated on the same DHT Server and reprovide them sequentially
- Periodically sweep the keyspace and reprovide CIDs

- Minimizing the number of connections to open and messages to send
- Number of connections to open $\approx$ number of DHT server peers

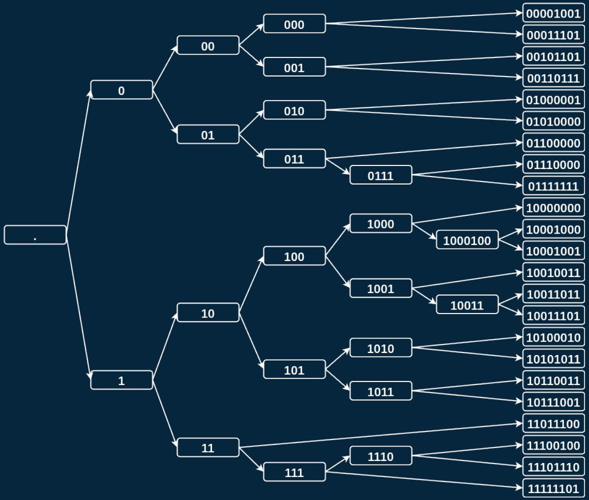
Binary Trie

- Binary Prefix Tree
- Data structure optimized for working with XOR distance
- Helps visualize locality in the Kademlia keyspace
- We will use them to group *close* CIDs



Keyspace Regions

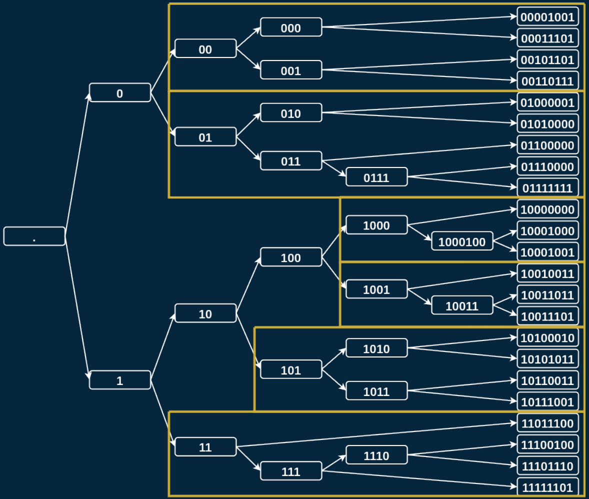
- A `Region` is defined as a prefix of the **peers** binary trie that has at least `repl` leaves.
- We are interested in the smallest possible regions fully covering the keyspace



Keyspace Regions: Example

Example: `repl = 3`

- Keys represent peers
- A `CID` is only stored in the region matching its prefix.



`Region` **exploration**

- Lookup a random key within the target `Region`
- If some returned peers don't match the `Region`'s prefix, `Region` is fully explored
- Else, take the largest fully explored subregion, and explore its neighbor subregion
- Repeat until the `Region` is fully explored

Sweep

- Explore Kademlia keyspace `Regions` to sweep the keyspace *from left to right*
- All Provider Records belonging to the looked up `Region` are reprovided
- The keyspace sweep takes `ReprovideInterval` to complete

- Peers are stored in a binary trie, in order to define the `Regions`
- CIDs are stored in a distinct binary trie, for fast sweep, insert and delete

Reproviding for a `Region`

- Define a temp key-value store `PeerID` → `[Keys]` for `PeerID`s within the `Region`
- For all keys `k` belonging to a `Region`, add `k` to its `repl` closest peers
- Iterate over all `PeerID`s within the `Region` and reprovide all associated keys
- The number of workers can easily be limited

`Region` **Reprovide Scheduling**

- Each `Region` should be republished within `ReprovideInterval` ± small delay
- Once a sweep cycle, the delays at which the `Regions` are reprovided are adjusted
- The Scheduler keeps track of the time the last reprovide happened for all `Regions`
- Enable resuming reprovide quickly if the node goes offline for some time

First Provide

- First Provide is timely, provide immediately
- Reprovide isn’t timely
- The first Reprovide of a CID is likely to happens less than `ReprovideInterval` after its first provide

`Region` **shrinking**

- A `Region` can shrink in the number of peers from one exploration to the next one
- A `Region` containing less than `repl` peers must be merged with its closest neighbor
- Providers Records are republished at the planned time, but the scheduler reschedule the reprovide time for the next round

Region **expansion**

- A Region can grow in the number of peers from one exploration to the next one
- A Region can be split in two Regions if both of its branches have at least repl peers
- Regions must always be splitted when possible
- When a Region is split in two (or more), the Provider Records associated with the new Regions are reprovided concurrently
- The scheduler is responsible to space the reprovide time for the next round

Intuition for proof of correctness

- First provides in appropriate `Region`, `Region` added to scheduler
- When a new CID must be provided, it is added to the appropriate `Region`
- `Regions` can be split or combined during peer churn within a `Region`

Performance Evaluation

Reproviding $1B$ CIDs to a $20'000$ peers network using `repl=20`.

- Number of `Regions`: $\sim \frac{\#peers}{2 \times repl} \approx 500$
- Number of connections opened to explore a `Region`: ~ 55
- Number of messages sent to explore a `Region`: ~ 70

- Total number of connections opened: $\sim 500 \times 55 \approx 28'000$
  - Vanilla Provide: $\sim 35B$, improvement: $\sim 1M\times$
- Total number of messages sent: $\sim 500 \times 70 + 20B \approx 20B$
  - Vanilla Provide: $\sim 70B$, improvement: $3.5\times$

Comparison with `ProvideMany` from the Accelerated DHT Client

Accelerated DHT Client `ProvideMany`:

+ Group CIDs by XOR distance before reproviding

- All CIDs are reprovided at the same time → rush hour
- No DHT lookup before providing → routing table stale entries
- Running a crawler to refresh the routing table → expensive
- Constant CIDs groups size → additional messages and connections

Migrating the Reprovide responsibility to the Content Routers

- The current DHT implementations only expose a `Provide(CID)` interface
- Currently IPFS implementations must handle the reprovide
- Different Content Routers have different reprovide mechanisms

- Each Content Router should be responsible to reprovide content
- Interface should be: `ProvideContent(CID)`, `UnprovideContent(CID)`, `GetProvidedContent()`

Conclusion

- Minimize Reprovide cost for everyone!
- Enable large Content Providers to use the DHT
- Once everyone uses the DHT, we can start moving away from the Bitswap broadcast
- To be shipped with the Double Hash DHT later this year

Reprovide Sweep Notion Page