Improving Lookup Latency in Distributed Hash Table systems Using Random Sampling

Presented by 曾胤煒
2005/11/11

IEEE/ACM TRANSACTIONS ON NETWORKING,
VOL. 13, NO. 5, OCTOBER 2005
Outline

- Introduction
  - DHT
  - Chord
- Lookup-Parasitic Random Sampling (LPRS)
- Simulation
- Conclusion
Distributed hash table (DHT)

- Provide routing infrastructures for scalable information storage and retrieval.
- Support scalable and distributed storage and retrieval of (key, data) pairs on the overlay network.
- Do this by associating each node in the network with a portion of the key space.
In most DHT systems, a request will take $\theta(\log N)$ overlay hops in average

(N : the network size)

A DHT network might have a large latency if the overlay network topology is not congruent with the underlying IP topology.

poor lookup performance
Chord

- Chord assigns each overlay node in the network an m-bit identifier (NodeID).
- Chord uses consistent hashing to assign keys to nodes.
- Each key is assigned to that node in the overlay whose node ID is equal to the key identifier, or follows it in the key space. That node is called the successor of the key.
successor(6) = 0

successor(1) = 1

successor(2) = 3
● Each node, $n$, maintains a finger table with $m$ entries.

● The $i^{th}$ entry contains the identity of the first node, $s$, that succeeds $n$ by at least $2^{i-1}$ on the identifier circle.
\[ S = \text{successor}(n + 2^{i-1}) \]
若 latency(0,4) > latency(0,5)

better?
● Each node must have a pointer to a good (i.e., low-latency) node in each geometrically decreasing range of the key-space.

● Simple way: to randomly sample a small number of nodes from this range.
Lookup-Parasitic Random Sampling

- **Range:** For a node $j$, its $i^{th}$ range is the interval of the key space defined as $[j+2^{i-1}, j+2^i]$

- **Sampling:** When we say node $x$ samples node $y$, we mean that $x$ measures the latency (e.g., by using ping) to $y$. 
LPRS

- When a request completes, each node on the request path samples the target, and updates its finger table accordingly.
If(latency(0,4) > latency(0,6))
Modification

- Each intermediate hop appends its IP address to a lookup message.
- A latency-sensitive replacement scheme to update its finger table entries based on the samples obtained.
Algo.

(1) A node $n$ maintains one finger table entry for each of its $m$ ranges.
(2) For node $p$, $n$ finds the range $k$ it lies in.
(3) If $p$ is closer than the current $k^{th}$ successor for node $n$, make $p$ the new $k^{th}$ successor of $n$. 
Simulation

- First phase: N nodes join the network. (We use the Chord join algorithms to build the finger tables.)
- Second phase: Nodes insert four documents into the network.
- Third phase: Each node generates, on average $3\log N$ request.
Hop-reach: the average latency on the underlying network incurred by each hop on the overlay network.
conclusion

- This paper describes LPRS, a fast random sampling technique

- Simulation based comparison with Chord establish the performance advantages of LPRS.
Thank you 😊