COEN 317 – Distributed System
Winter 2020

NIUBILITY:

A Chord Based Distributed Hash Table

Proposed by,

Zhulian Huang (W1280699)
Jiabei Luo (W1142195)
Yu Li (W1549505)
Zihan Zhang (W1538040)
Date: February 7, 2020

Under the guidance of Prof. Dr. Ramin Moazzeni

I. Rationale for the project

Peer-to-peer systems provide us with many benefits including avoidance of single point of failure by a centralized server, flexible joins and leaves, and hardware cost efficiency. However, there are also many complex issues associated with designing and building a robust, performant and scalable distributed peer-to-peer system. Our project examines the most common problems, such as lookup performance, large concurrent joins/leaves, and fault-tolerance. We attempt to address those problems with algorithms we have closely analyzed and found to be suitable for our specific application, which is a Distributed Hash Table (DHT).

Algorithm Taxonomy	Structured P2P Overlay Network Comparisons			
	CAN	Chord	Tapestry	Pastry
Decentralization	DHT functionality on Internet-like scale			
Architecture	Multi-dimensional ID coordinate space.	Uni-directional and Circular NodeID space	Plaxton-style global mesh network	Plaxton-style global mesh network
Lookup Protocol	Key, value pairs to map a point P in the coordinate space using uniform hash function.	Matching Key and NodeID.	Matching suffix in NodeID	Matching Key and prefix in NodeID
System Parameters	N-number of peers in network d-number of dimensions.	N-number of peers in network	N-number of peers in network B-base of the chosen peer identifier	N-number of peers in network b-number of bits (B = 2^b) used for the base of the chosen identifier
Routing Performance	$o(d.N^{\frac{1}{d}})$	$o(logN)$	$O(logB^N)$	$O(logB^N)$
Routing State	2d	$logN$	$logB^N$	$BlogB^N + BlogB^N$
Peers Join/Leave	2d	$(logN)^2$	$logB^N$	$logB^N$
Security	Low level. Suffers from man-in-middle and Trojan attacks			
Reliability/Fault Resiliency	Failure of peers will not cause network- wide failure. Multiple peers responsible for each data item. On failure application retries.	Failure of peers will not cause network-wide failure. Replicate data on multiple consecutive peer. On failures, application retries.	Failure of peers will not cause network-wide failure. Replicate data across multiple peers. Keep track of multiple paths to each peer.	Failure of peers will not cause network-wide failure. Replicate data across multiple peers. Keep track of multiple paths to each peer.

Table 1 A Comparison of Various Structured P2P Overlay Network Schemes [4]

A Distributed Hash Table (DHT) helps searching for a file efficiently with a keyword in peer-to-peer (P2P) networks. The DHT-based P2P networks are structured P2P networks. On the contrary, unstructured P2P networks use flooding or a central server for search. Structured P2P networks offer more scalable, efficient, and robust search and management. Therefore, our team focuses on structured P2P networks, such as Content-Addressable-Network(CAN), Chord [1], Tapestry and Pastry. *Table 1* is a summary of these structured P2P overlay network schemes compared from 8 aspects: architecture, lookup protocol, system parameters, routing performance, routing state, peers join/leave, security and reliability/fault resilience.

Based the comparison, our team is interested in implementing a DHT using Chord protocol because Chord simplifies the design of peer-to-peer systems and applications based on it by addressing these difficult problems: Load Balance, Decentralization, Scalability, Availability and Flexible naming.

For example, Chord's design is substantially simpler and handles concurrent node joins and failures well. The Chord protocol is also similar to Pastry, the location algorithm used in PAST. However, Pastry is a prefix-based routing protocol, and differs in other details from Chord. The state maintained by a Chord node does not depend on the network size, and the lookup cost is logN. However, the lookup cost of a CAN node increases faster than logN. Chord also has the advantage that its correctness is robust in the face of partially incorrect routing information.

Besides a literature review on different types of DHT architecture, we will also gain hands-on experience in building a DHT with necessary features from scratch. The experience of implementing a complex system will be significantly helpful in our careers both academically and professionally.

II. Specific goals of the project

The goal of our project is as follows [1]:

A. Implementing a Distributed Hash Table (DHT) which is a structured peer-to-peer system based on Chord protocol.
B. Using consistent SHA-1 hashing to give identifier to string and socket address.
C. The DHT could deal with lookup, concurrently nodes join, nodes fail or leave voluntarily.
D. The DHT would incorporate several classes of distributed algorithms such as,
   1) Concurrency control by using multi-threading programming
   2) Resource discovery by using scalable key location mechanism using figure table to realize *O(logN)* routing performance
   3) Replication and consistency by using stabilization protocol

III. Challenges

The difficulties we anticipate to face in developing this project are [1]:

A. Scalability: How do we make sure the performance of joins/leaves/updates/lookups isn't impacted when the network becomes much larger and the structure of the network is constantly updated?
B. Availability/Robustness: How do we ensure that when nodes go down, data in the finger tables are not lost and lookup correctness is maintained?
C. Decentralization: How do we ensure that the consistency of replicated data is maintained without having a centralized manager/server node?

The above challenges are all addressed by the Chord protocol. Chord uses SHA-1 to ensure load balancing properties, automatically updates the finger tables to ensure a certain key can always be found and maintains log(N) lookups as long as finger tables are correct.

IV. A detailed plan

Date	Tasks
1/26/2020 - 2/1/2020	Research several distributed system applications and determine the goals of the project.
2/2/2020 - 2/8/2020	Determine the algorithms, correctness, and complexity involved in implementing a structured peer-to-peer system based on Chord protocol.
2/9/2020 - 2/15/2020	Draft report abstract and introduction. Elaborate on the idea of each algorithm and method, including concurrency control, resource discovery, scalability, performance, etc..
2/15/2020 - 2/21/2020	Implement detailed functions in DHT. Compare performance, protocols, and usages of each algorithm.
2/22/2020 - 2/29/2020	Complete algorithms and implementation. Write test cases and analyze the expected performance. Collect the failure during the project. Draft report conclusion.
3/1/2020 - 3/7/2020	Achieve the goals of the project. Record a video demo. Create slides for the project presentation. Prepare for the presentation
3/8/2020 - 3/13/2020	Accomplish the representation and submit code, test case and a final report.

V. Reference:

[1] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. Tech. Rep. TR-819, MIT LCS, March 2001. http://www.pdos.lcs.mit.edu/chord/papers/.

[2] KARGER, D., LEHMAN, E., LEIGHTON, F., LEVINE, M., LEWIN, D., AND PANIGRAHY, R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (El Paso, TX, May 1997), pp. 654–663.

[3] LEWIN, D. Consistent hashing and random trees: Algorithms for caching in distributed networks. Master's thesis, Department of EECS, MIT, 1998. Available at the MIT Library, http://thesis.mit.edu/.

[4] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim , A Survey and Comparison of Peer-to-Peer Overlay Network Schemes, IEEE COMMUNICATIONS SURVEY AND TUTORIAL, MARCH 2004

[5] Stephanos Androutsellis-Theotokis, White Paper: A Survey of Peer-to-Peer File Sharing Technologies, Athens University of Economics and Business, The eBusiness Centre