THE DESIGN OF RESILIENCE P2P NETWORKS WITH DISTRIBUTED HASH TABLES

Dorina Luminiţa COPACI, Constantin Alin COPACI
lcopaci@yahoo.com, acopaci@yahoo.com

Abstract

The term resilience in computer systems and networks is often used in relationship with other terms, e.g., fault tolerance, dependability, availability, reliability. Hash tables are important in P2P networks. In this paper, we discuss about two concepts: a Content-Addressable Network (CAN) and Chord as distributed infrastructures that provides hash table on Internet.

Keywords: Peer-to-peer networks, resilience, distributed hash table

1. Introduction

Peer-to-peer (P2P) networks [18] are generally overlays on top of existing networks, which allow for functionality not provided by the underlay. In contrast to client/server systems with asymmetric roles, each peer in a P2P network implements the functions of both client and server.

Many peer-to-peer networks [25], [28], [33], [37] are based on *distributed hash tables* (DHTs), which provide a decentralized, many-to-one mapping between user objects and peers. Their distributed structure, excellent scalability, short routing distances, and failure resilience make DHTs highly suitable for peer-to-peer networks.

This article is organized as follows: section 2 presents algorithmic and operational aspects for distributed hash table and section 3 summarizes the main findings of the article. In this article are described the Chord and CAN algorithms for analyzing the resilience, using DHTs, of P2P networks.

2. Distributed Hash Tables

Distributed Hash Tables (DHTs) [12], [21], [22] use a network overlay to partition a set of keys among participating nodes, and can efficiently route messages to the unique owner of any given key.

2.1. Algorithmic aspects

In this section, we consider structured overlays in an abstract way. We basically consider them as graphs, and approach their algorithmic aspects, which are mainly related to the topology how the overlay is built the routing algorithm. We will present the Chord and CAN algorithms for analyzing the resilience of P2P networks.

2.1.1. The Chord algorithm

The Chord [7] only support the mapping of a key onto a node, storing a value associated with the key is left to the application layer. It also guarantees that queries make a logarithmic number of hops. The consistent hashing assigns each node and key a n-bit identifier using a base hash function. Identifiers are ordered in an identifier circle modulo 2^n [7]. Key k is assigned to the first node whose identifier is equal or follows k in the identifier space; this node is called the successor node of key k. Each Chord node only maintains a small total of "routing" information about other nodes.

A node recognize and keeps information about: its successor on the circle; a finger table whose i^{th} entry (where $1 \le i \le n$) contains the identity of the first node that succeeds N by at least 2^{i-1}; a predecessor pointer which contains the Chord identifier and IP address of the immediate predecessor of the node.

In a network with N nodes, each node maintains information only about O(logN) other nodes and a lookup is done in O(logN) hops. When a node joins or leaves the network, Chord must update the routing information. These operations need $O(\log_2 N)$ messages.

To store the Chord network integrity while nodes join and leave, it needs to store two invariants: each node's successor is correctly maintained; for every key k, node *successor(k)* is responsible for *k*.

When a node *n* joins the network, Chord must execute three tasks to store the invariants:

1. Initialize the predecessor and fingers of node n.
2. Update the fingers and predecessors of existing nodes to reflect the addition of n.
3. Notify the higher layer software so that it can transfer state (e.g. values) associated with keys that node n is now responsible for.

We consider a small Chord overlap as in Figure 2.1.1., where m= 8 (m is number of inputs in finger table).

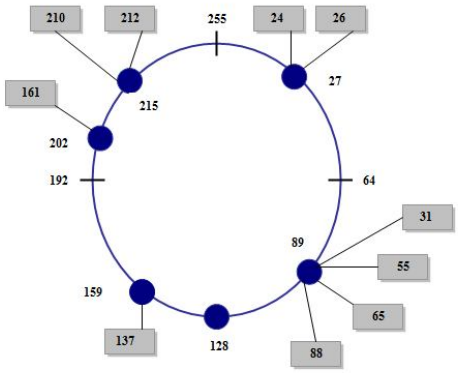


Figure 2.1.1. Overlap Chord with m=8

The Chord finger table for the 27 node is shown in Table 2.1.1.:

k	The nearest node of	The actual node
0	2^0+27	89
1	2^1+27	89
2	2^2+27	89
3	2^3+27	89
4	2^4+27	89
5	2^5+27	89
6	2^6+27	128
7	2^7+27	159

Table 2.1.1. The Chord finger table for 27 node

In this example, the 27 node wants to find content whose location in the overlap is 210. The node consults the finger table and found that the nearest node is 159. The 159 node queries the 27 node on location of the 210 node. The 159 consults the finger table and returns 202. Then, 27 contacts 202 which return 215. Then 27 contacts 215 which indicate that it is responsible for 210. At this point, 27 queries 215 to find the information what it wants. This type of routing is known as iterative routing, where a number of iterations or steps are made from source to destination. The source is always in control and can check the integrity of routing.

Another routing approach is recursive approach, where demand is around coverage hop until it reaches its destination. This is faster than iterative routing and implies less processing on the source.

To deal with simultaneous joins and leaves, Chord uses a "stabilization" protocol to keep nodes' successor pointers up to date. The stabilization process is run periodically by every node.

Chord itself does not deliver any authentication and replication mechanisms. The application layer is responsible for delivering such mechanisms if needed.

2.1.2. The Content Addressable Network (CAN) algorithm

A Content Addressable Network (CAN) [8], [9], [10] is built around a virtual multi-dimensional Cartesian coordinate space on a multi-torus. The entire coordinate space is dynamically partitioned among all the peers in the system such that every peer has its individual zone within the total space. A CAN peer retains a routing table that has the IP address and virtual coordinate zone of each of its neighbor coordinates. A peer routes a message to its destination using a simple greedy forwarding to the neighbor peer that is closest to the destination coordinates [25]. A CAN involves an additional maintenance protocol to periodically remap the identifier space onto nodes.

2.1.3. Consistent Hashing

Consistent hashing has been introduced in [65]. It aims for each peer receiving approximately the same number of items to be stored. It escapes the necessity of re-distributing the complete hash table each time a new machine joins the network. Any change in the network, e.g., node joins and leave, requires only adjustments at a local scope. Consistent hashing is generally useful where multiple machines with different views of the network need to agree on common tasks without communication.

2.1.4. Geometry

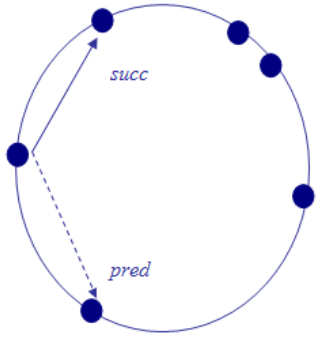
Algorithms for Distributed Hash Tables can be classified according to the geometry used for their presentation (e.g., rings, trees and hypercubes). The major design criteria are generally:

1. Lookup performance,
2. Routing table size,
3. Fault tolerance.

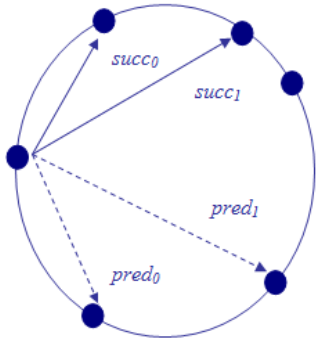
These design criteria are strongly interrelated. In a full mesh overlay with a routing table size in O(n) (design criterion 2.) any node can reach any other node with one hop (design criterion 1.). An additional advantage of this topology is that the graph is tolerant against link and node failures (design criterion 3.). Nevertheless, the problem with such overlays is that the routing table size is not scalable in many cases. Note that while the memory required to store a routing table may not be of a problem even with a couple of million peers.

Another example of a structured overlay is shown in Figure 2.1.4. (a). Every node is connected to its successor and predecessor on a ring. Despite of being simple, a further advantage of this topology is that every node requires only maintaining a link to two other nodes (successor and predecessor) (design criterion 2.). However, the drawbacks are that:

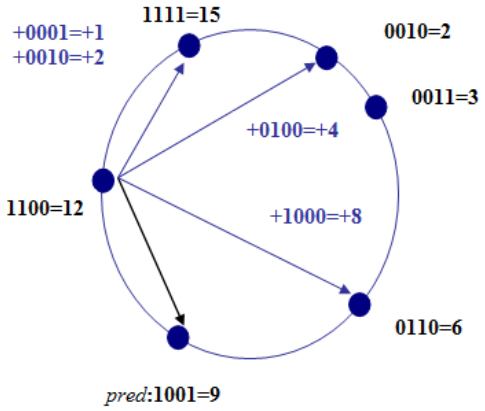
- Lookups take *O(n)* (design criterion 1.),
- The topology is not sufficiently fault-tolerant (design criterion 3.). The failure of only two nodes leads to a partitioned network.



(a) Simple ring: single succ and single pred for each node



(b) Ring with multiple succ and pred for each node



(c) Chord ring

Figure 2.1.4. Ring with succ. and pred. for each node

2.2. Operational Aspects

Since peers join and leave the network, the design of a structured overlay needs to accommodate for the so called churn. Furthermore, the routing performance can be improved in terms of latency and success rate based on several optimizations discussed below. Additional aspects are connectivity across middle boxes and heterogeneous node capacities which have often lead to the differentiation between super peers and other nodes in practice. In the following subsections, we provide a brief overview of these issues which need to be considered when designing and building P2P networks.

We concentrate on classic vulnerabilities of Distributed Hash Tables and we discuss how to prevent them:

1. *Authentication* in a Distributed Hash Table without introducing some form of centralization is a real challenge because Distributed Hash Table is decentralized and authentication not being really possible without some sort of centralized credentials server.
2. *Rogue node*: the node can either modify or delete on its own the information that it stores, because the information stored in the DHT is distributed among the participating nodes.
3. *Denial of service*: the attacker can perform a Denial of Service (DoS) attack by inserting a lot of successive faulty nodes into the system. These inserted nodes will stock all the pointers maintained by the node preceding them in the overlay cutting off the targeted nodes from the DHT.
4. *Sybil attack*: The Sybil attack [18] consists in controlling a substantial fraction of the overlay by inserting a lot of faulty nodes operated by a single entity in the system. The attacker can perform different kinds of attack by controlling a large amount of nodes.

2.2.1. Protections against attacks

Protection from rogue nodes

Encrypting exchanged data can ensure their integrity but does not stop a node to return an empty result or to report the data as missing from the DHT. Replication techniques can add some protection against such comportment. Performing a check of the node during its engagement within the overlay can increase the security or using a reliability network for nodes known from the DHT, but what if the node gets corrupted?

Protecting from Sybil attacks

In [18], John R. Douceur states that “without a logically centralized authority, Sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities”. Protection against Sybil attacks in a fully decentralized DHT seems therefore impossible.

3. Conclusion

In this paper, we provided two algorithms for designing of P2P networks with a special focus on DHTs. We separated between algorithmic aspects where P2P networks are considered from a graph theoretic perspective and operational aspects where deployment considerations are taken into account.

REFERENCES

[1] J. Aspnes, Z. Diamadi, and G. Shah, "Fault-Tolerant Routing in Peerto-Peer Systems," *ACM PODC*, July 2002.

[2] Y. Azar, A. Broder, A. Karlin, and E. Upfal, "Balanced Allocations," *SIAM J. on Computing*, vol. 29, no. 1, 1999.

[3] A.-L. Barabasi, R. Albert, and H. Jeong, "Scale-free Characteristics of Random Networks: The Topology of the World Wide Web," *Physica* A **281**, 2000.

[4] T. Bu and D. Towsley, "On Distinguishing between Internet Power Law Topology Generators," *IEEE INFOCOM*, 2002.

[5] C. Baransel, W. Doboseiwicz, and P. Gburzynski, "Routing in Multi-hop Packet Switching Networks: Gbps Challenge," *IEEE Network Magazine*, 1995.

[6] W.G. Bridges and S. Toueg, "On the Impossibility of Directed Moore Graphs," *Journal of Combinatorial Theory*, series B29, no. 3, 1980.

[7] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-to-Peer lookup service for internet. In ACM SIGCOMM 2001, San Diego CA, August 2001. [9] J. Considine and T.A. Florio, "Scalable Peer-to-Peer Indexing with Constant State," *Boston U. Technical Report 2002-026*, August 2002.

[8] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable Content-Addressable Network. In ACM SIGCOMM 2001, San Diego, CA (USA), August 2001.

[9] A. Fiat and J. Saia, "Censorship Resistant Peer-to-Peer Content Addressable Networks," *Symposium on Discrete Algorithms*, 2002.

[10] P. Fraigniaud and P. Gauron, "An Overview of the Content-Addressable Network D2B," *ACM PODC*, 2003.

[11] P. Ganesan, Q. Sun, and H. Garcia-Molina, "YAPPERS: A Peer-to-Peer Lookup Service over Arbitrary Topology," *IEEE INFOCOM*, 2003.

[12] K.P. Gummadi, R. Gummadi, S.D. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The Impact of DHT Routing Geometry on Resilience and Proximity," *ACM SIGCOMM*, August 2003.

[13] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. on Information Theory*, March 2000.

[14] K. Hildrum, J. Kubiatowicz, S. Rao, and B.Y. Zhao, "Distributed Object Location in a Dynamic Network," *ACM SPAA*, August 2002.

[15] F. Kaashoek and D.R. Karger, "Koorde: A Simple Degree-optimal Hash Table," *IPTPS*, February 2003.

[16] C. Law and K.-Y. Siu, "Distributed Construction of Random Expander Graphs," *IEEE INFOCOM*, 2003.

[17] F.T. Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes," *Academic Press / Morgan Kaufmann*, 1991.

[18] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the Evolution of Peer-to-Peer Networks," *ACM PODC*, 2002.

[19] D. Loguinov, "Evolution of Massive P2P Graphs: Zone Distribution Perspective," *Work in Progress*, July 2003.

[20 D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience," *ACM SIGCOMM*, August 2003.

[21] G.S. Manku, **"**Routing Networks for Distributed Hash Tables," *ACM PODC*, June 2003.

[22] G.S. Manku, M. Bawa, and P. Raghavan, "Symphony: Distributed Hashing in a Small World," *USITS*, 2003.

[23] M. Naor and U. Wieder, "Novel Architectures for P2P Applications: the Continuous-Discrete Approach," *ACM SPAA*, June 2003.

[24] G. Pandurangan, P. Raghavan, and E. Upfal, "Building Low-Diameter P2P Networks," *IEEE Symposium on Foundations in Comp. Sci.*, 2001.

[25] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *ACM SIGCOMM*, 2001.

[26] S. Ratnasamy, S. Shenker, I. Stoica, "Routing Algorithms for DHTs: Some Open Questions," *IPTPS*, 2002.

[27] S.M. Reddy, J.G. Kuhl, S.H. Hosseini, and H. Lee, "On Digraph with Minimum Diameter and Maximum Connectivity," *Proceedings of Allerton Conf. on Communications, Control and Computers*, 1982.

[28] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems," *IFIP/ACM International Conference on Distributed Systems Platforms*, 2001.

[29 J. Douceur. The Sybil attack. In Proc. of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002

[30] J. Saia, A. Fiat, S. Gribble, A.R. Karlin, and S. Saroiu, "Dynamically Fault-Tolerant Content Addressable Networks," *IPTPS*, March 2002.

[31] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl, "HyperCuP – Hypercubes, Ontologies and Efficient Search on P2P Networks," *Workshop on Agents and P2P Computing*, 2002.
[32] Sean Christopher Rhea. OpenDHT: A Public DHT Service. PhD thesis, University of California, Berkeley, 2005. Data Dissemination," *ACM NOSSDAV*, June 2001.
[33] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *ACM SIGCOMM*, August 2001.
[34] D.A. Tran, K.A. Hua, and T.T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," *IEEE INFOCOM*, 2003.
[35] J. Xu, "Topological Structure and Analysis of Interconnection Networks," *Kluwer Academic Publishers*, 2002.
[36] J. Xu, A. Kumar, and X. Yu, "On the Fundamental Tradeoffs between Routing Table Size and Network Diameter in Peer-to-Peer Networks," *To Appear in IEEE JSAC*, November 2003.
[37] B.Y. Zhao, J.D. Kubiatowicz, and A. Joseph, "Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing," *UC Berkeley Technical Report*, April 2001.
[38] S.Q. Zhuang, B.Y. Zhao, and A.D. Joseph, "Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area