Distributed Hash Tables

Aaron Kaluszka

IS250

15 April 2010

The distributed hash table is a type of distributed and decentralized system that provides a lookup service similar to a hash table. Hash tables are data structures that use a hash function to efficiently map identifiers known as keys to associated values (e.g. an individual's name to his/her phone number). A hash function allows you to perform a trivial calculation, which will then direct you to the data rather than needing to look through an entire database. Most distributed hash tables use Secure Hash Algorithm 1 (SHA-1), which yields a 160-bit key.

In a distributed hash table, any participating node can quickly retrieve data based on keys. Rather than a single central server, responsibility for this mapping is distributed among the nodes, and is done so in such a way that changes in which nodes are connected to the system cause minimal disruption. Due to this design, distributed hash tables can support extremely large numbers of nodes, while handling continual arrivals, departures, and failures.

The data stored in the hash table may not be the actual data requested; it may simply be a link to the data. For instance, when using a distributed hash table as part of a search engine, the result returned from the system may be a link to the server hosting the content searched for rather than the content itself.

Distributed hash tables include a keyspace, the set of all possible keys, which is split up across the nodes in the system. To map keys, another function is defined that describes the distance from one key to another. Each node is assigned an identifier, which is itself a key and the node owns all keys and associated data within that given distance. This means that if a node is removed from the network, only a small portion of the data must be recovered by other nodes.

Each node maintains a list of links to other nodes, effectively a routing table. In this way, they act as a virtual network overlay that resides on top of the physical network. In the overlay network, nodes are connected sparsely and a key lookup is passed from node to node until it reaches the node with a keyspace covering that key. For any key, a node either owns the key or has a link to a node that is closer to the key. This setup makes it easy to route the message using a greedy algorithm; each node simply passes the request to the node that is closest to the key based on the distance measure. This is accomplished in different ways depending on the specific implementation. One popular design, Kademlia, is based on a tree structure with branches split up based on the leading bits of the key. Another system, Chord, uses points on a circle to define the keyspace.

To reduce the amount of traffic required to obtain results and thus make the routing efficient, the maximum number of neighbors any node maintains is low, and the maximum route length is also kept low. Both of these parameters are often set at log *n*, where *n* is the total number of nodes. Some systems may keep a record of routing paths to increase efficiency and to provide network self-healing. This design also increases resistance against denial of service attacks since a limited number of nodes will be affected, making the system effectively impossible to shut down.

Distributed hash tables reside in the application layer, though they are not usually the applications themselves. They are used in complex applications such as distributed file systems, multicast, domain name systems, and instant messaging. One of their most famous uses is in peer-to-peer file sharing; Bittorrent, eDonkey, and newer versions of Gnutella use distributed hash tables in conjunction with more traditional means of data distribution. They are also used in content distribution systems and cooperative web caching. For example, the Coral Content Distribution Network caches websites that are hit with large amounts of traffic. There are a number of experimental search engines such as YaCy, which use distributed hash tables to index content. They are also used in anonymous networks such as Freenet, which can be used to bypass censorship by passing information indirectly through a number of nodes.

Distributed hash tables provide a fast and fault-tolerant way of accessing large amounts of data with minimal cost.

References

"Distributed Hash Table." *Wikipedia*. http://en.wikipedia.org/wiki/Distributed_hash_table

Gribble, Steven D., *et. al.* "Scalable, Distributed Data Structures for Internet Service Construction." *4th Symposium on Operating System Design & Implementation*. 25 Oct 2005. San Diego, CA. http://www.usenix.org/events/osdi00/gribble.html

"Hash Table." *Wikipedia*. http://en.wikipedia.org/wiki/Hash_table

Maymounkov, Petar and David Mazieres. "Kademlia: A Peer-to-peer Information System Based on the XOR Metric." *Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems*. 07 Mar 2002. MIT Faculty Club, Cambridge, MA. http://www.cs.rice.edu/Conferences/IPTPS02/109.pdf

Stoica, Ion, *et. al.* "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications." *Transactions on Networking.* 11(1). Feb 2003. http://pdos.csail.mit.edu/papers/ton:chord/

Wiley, Brandon. "Distributed Hash Tables." *Linux Journal*. 01 Oct 2003. http://www.linuxjournal.com/article/6797