DISTRIBUTED HASH TABLES

A distributed hash table (DHT) is a type of distributed system that provides a lookup service similar to a hash table. In a hash table, data is stored and retrieved using keys, and the keys are used to determine the location of the data in the table. A distributed hash table is similar, but the data is distributed across multiple nodes in a network rather than being stored in a single table.

In a DHT, each node is responsible for storing and managing a portion of the data. When a client wants to retrieve or store data, it sends a request to the network. The request is then forwarded to the appropriate node based on the key of the data being requested. The node then responds to the request and either retrieves or stores the data.

DHTs are used in a variety of applications, including peer-to-peer (P2P) networks, distributed databases, and distributed file systems. They are particularly useful for large-scale distributed systems, as they provide an efficient and scalable way to store and retrieve data.

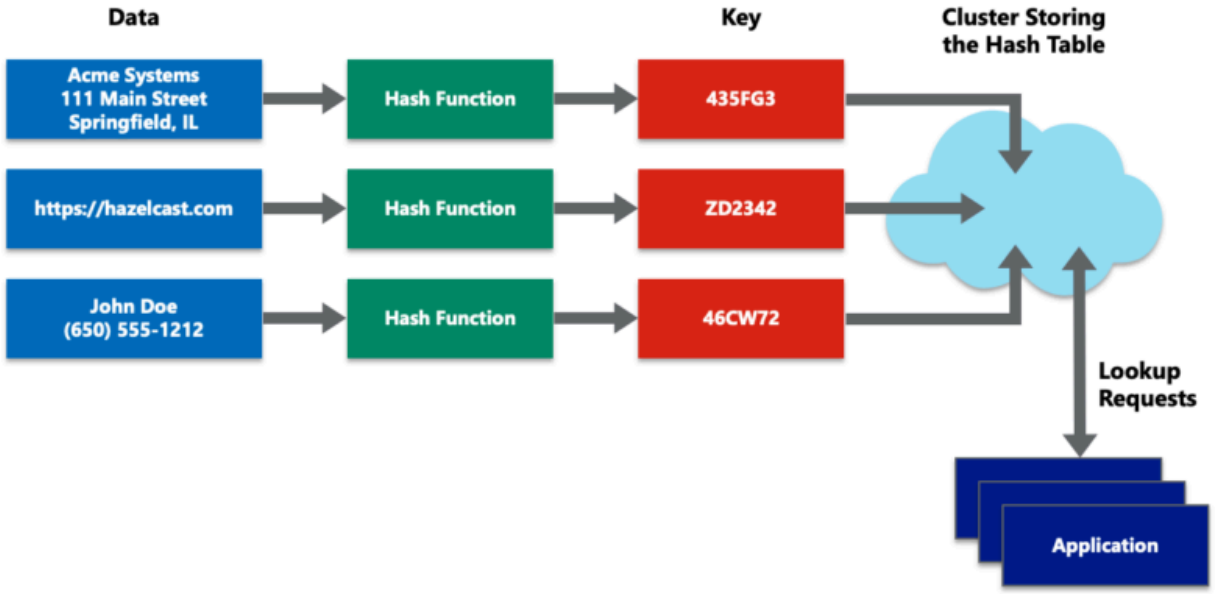
DHTs are used for a variety of purposes, including –

- ➔ Peer-to-peer (P2P) networks – DHTs are often used in P2P networks to facilitate the sharing of resources, such as files or data, between peers. DHTs allow peers to locate resources on the network and download them directly from one another.

- ➔ Distributed databases – DHTs can be used to store and retrieve data in a distributed database. Because the data is distributed across multiple nodes in the network, DHTs can provide a scalable and efficient way to store and retrieve large amounts of data.

- ➔ Distributed file systems – DHTs can be used to store and manage files in a distributed file system. By distributing the files across multiple nodes, DHTs can provide a scalable and fault-tolerant way to store and access large amounts of data.

- ➔ Content delivery networks – DHTs can be used to store and distribute content, such as videos or images, across a network of servers. This can help to reduce the load on a single server and improve the performance of the network.



Structure

The structure of a DHT can be decomposed into several main components. The foundation is an abstract keyspace, such as the set of 160-bit strings. A keyspace partitioning scheme splits ownership of this keyspace among the participating nodes. An overlay network then connects the nodes, allowing them to find the owner of any given key in the keyspace.

Once these components are in place, a typical use of the DHT for storage and retrieval might proceed as follows. Suppose the keyspace is the set of 160-bit strings. To index a file with given filename and data in the DHT, the SHA-1 hash of filename is generated, producing a 160-bit key k, and a message put(k, data) is sent to any node participating in the DHT. The message is forwarded from node to node through the overlay network until it reaches the single node responsible for key k as specified by the keyspace partitioning. That node then stores the key and the data. Any other client can then retrieve the contents of the file by again hashing filename to produce k and asking any DHT node to find the data associated with k with a message get(k). The message will again be routed through the overlay to the node responsible for k, which will reply with the stored data.