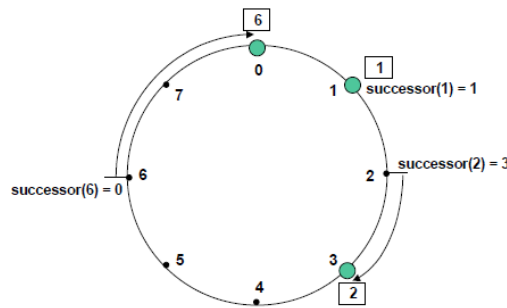


Paper Review

“Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”

1. Summary

When implementing a distributed protocol, it could be hard to change the number of nodes in the system. There are difficulties when a new node joins the system or a node leaves the system because we need to give the new node something to store or put data to other working nodes. In traditional distributed systems, we need to calculate all the hashes which are costing. In chord, the complexity of joining or leaving is $O(\log^2 N)$ where N is the number of nodes in the system because it limits the number of elements we need to compute and update.



Chord will calculate the tag of each node by hashing and insert this node to a ring by its hash number shown as 1. For any resource r , it will be placed in the first node on the ring clockwise right to $hash(r)$. The insert and leave of nodes can cause a limited impact on the whole system.

Each node will maintain a finger table of the router table of the current node, the successor, and the predecessor. A new data structure called Interval was maintained in each node acting as a router. Each node has $\log_2(digitsofhash)$ Intervals, for example a 4-bit-hash has 4 Intervals, $[0, 1)$, $[1, 2)$, $[2, 4)$, $[4, 8)$, and each Interval will record the last one node to the Interval. (The last one means the next node can see on this ring clockwise). Such Intervals act as a routing table and saves time for a distributed hash table because if one node wants to find another node, it can first look up a node's Interval and a node's Interval's Interval with a complexity of $O(\log N)$.

When inserting a new node or deleting a node, a few things need to be updated. The successor of each node, the resource to store, finger tables, and Intervals.

2. Advantages

- + When there has a change happened in the system, the complexity of change is low and limited.

- + Good scalability. Two chord with the same hash function can easily merge into one chord by add a digit to the hash.
- + Distributed routing table, which contributes to great robustness.

3. Disadvantages

- Insecurity: the chord can be seen as a linked list, and if one node is online but refused to cooperate, the link list will be hard to process because the successor and predecessor cannot offer valid information.
- Lack of optimization of physical networks.
- A new node still needs to achieve information from $\log^2 N$ nodes, which is still costly for large systems, especially network failures happens.