Report HW5: Chordy - a distributed hash table

Artem Sliusarenko

October 11, 2023

1 Introduction

The Base *chord protocol* is responsible for key lookup, new nodes joining the system, and handling node failures. *Chord protocol* allows key lookup in $log(N)$ messages to other nodes and does not require each node storing routing information to all other nodes in the system. Chord must keep the routing infotmation up tp date. Stabilizing the routing table when nodes leave or join the system takes $log^2(N)$ messages.

2 Main problems and solutions

3 Evaluation

Stabilization process is responsible for including new nodes into the ring. If we don't do the stabilization more frequently key-value lookups performance may decrease or fail ocasionally. Less frequent stabilization procedure delays the update of nodes' successor pointers. The cons of the more frequent stabilization procedure is the $log^2(N)$ of messages going through the system every time it's triggered. First version of the node module is capable of adding new nodes to the system. We can send the probe through the system. It will go through every node and print the time it took to reach probe sender node. We also have possibility to view each nodes successor and predecessor using the status message.

Performance testing.

1. Running the ring with only 1 machine. 4 machines are tasked to add 1000 elements and perform lookup. Each machine completed the task withing approximately 40 ms.

2. Running the ring with 4 machines. 4 machines are tasked to add 1000 elements and perform lookup. Each machine completed the task withing approximately 40 ms.

It does not take longer for 1 machine to handle 4000 elements then 4 machines handle 1000 elements each. It does not matter how many machines we add at random or which machine will receive initial lookup request. Chord is a decentrilized system. All key lookup requests are resolved in $log(N)$ messages to other nodes.

4 Conclusions

Chord is a decentralized system that can resolve key lookups in $log(N)$ messages where N is the number of machines in the system. Frequent stabilization procedure makes sure the routing table is up to date to avoid degraded performcance or key lookup failure.