Distributed Operating System Principles Project - 3

Kavya Gopal Rema Veeranna Gowda

3rd November 2021

1 Problem Statement

The goal of this project is to simulate a peer to peer network. We need to implement chord protocol in F# using actor model.

2 Algorithm

Chord Protocol

Chord is one of the protocol for a peer to peer distributed system. We create a ring structure with all nodes that are sparsely distributed. Each node has a finger table that has key-value pairs that has information on the nodes having information. This reduces the search time for a key in node from O(n) to O(log n).

- Create chord ring: We place all nodes in the form of a ring. Each time a node enters or leaves the distributed system, we update the ring.
- Finger Table: Each node maintains a finger table containing m entries(number of bits in hash key). The first entry in the table is the immediate successor of the node, the remaining entries have node that are further away from the current node in a logarithmic way. The finger table is maintained when ever a node enters or leaves the distributed system.
- When a node joins or leaves the distributed system: Each node has a successor that points to the next node in the ring and a predecessor that points to the previous node. Each time a node enters, we update other nodes to update their predecessor, successor and finger tables. The new node takes required keys from its successor.
- Search for a node that contains Key: We use finger tables to find the node that would contain the key. We hop on to the next node based on the values in the finger table. The average number of hops is calculated in this implementation.

3 Solution

Implementation Details

Chord protocol is working for the following functionalities:

1. Join nodes to network.
2. Message transfer among nodes in network.
3. We are updating finger table.
4. Stabilize the chord network whenever a node is added.
5. We are calculating the average number of hops.

Running the code

To run the program, we need to provide 2 inputs:

dotnet fsi –langversion:preview project3.fsx numOfNodes noOfRequests

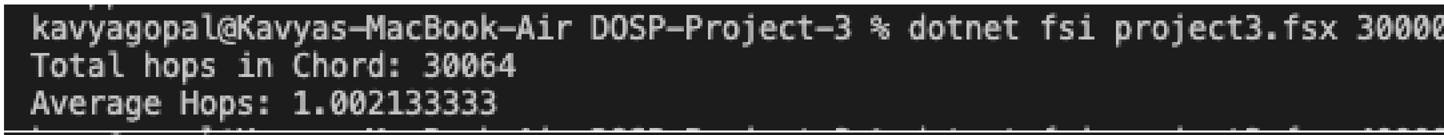
- numOfNodes: number of nodes for the chord ring
- noOfRequests: number of keys each actor searches far

Output

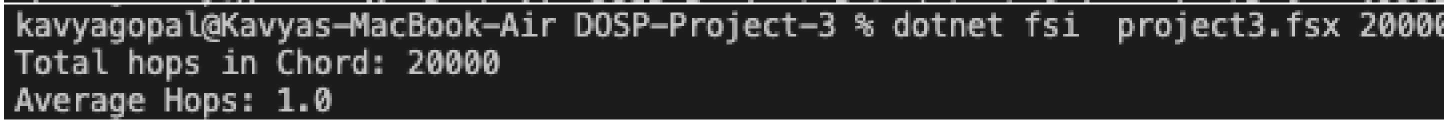
***Average Hops**: Average number of hops that were taken to search keys in the chord network*

***Termination** - The program is terminated when all the actors successfully search 'noOfRequests' number of keys.*

***Largest network we managed to run code**: 30000 nodes*



```
kavyagopal@Kavyas-MacBook-Air DOSP-Project-3 % dotnet fsi project3.fsx 30000
Total hops in Chord: 30064
Average Hops: 1.002133333
```



```
kavyagopal@Kavyas-MacBook-Air DOSP-Project-3 % dotnet fsi  project3.fsx 20000
Total hops in Chord: 20000
Average Hops: 1.0
```