COP5615 – Distributed Operating Systems

Project Report – 3

Team Members

1. Akshay Ganapathy (UFID - 3684-6922)
2. Kamal Sai Raj Kuncha (UFID - 4854-8114)

Input

The input to the program is the number of nodes (the number of nodes used to create the chord peer to peer system), and the number of requests (the number of requests each node makes).

Output

The output is the average number of hops that have been traversed in order to deliver the message.

Zip File Contents

The zip file consists of the readme.md file, Project_Report.pdf file, and the project3.fsx file which contains the code to be run.

How To Run

Run the project3.fsx file using the command:
“dotnet fsi project3.fsx <numNodes> <numRequests>”

where ‘numNodes’ is the number of nodes used to create the chord peer to peer system, and ‘numRequests’ is the number of requests each node makes.

Languages used

F# was used to code the project.

Platforms used for running the code

Visual Studio Code
.NET version 5.0
NuGET Akka.NET v1.4.25

Largest Network Obtained

The largest network we were able to work with was 5000 nodes and 10 requests. The average number of hops was found to be 5.846.

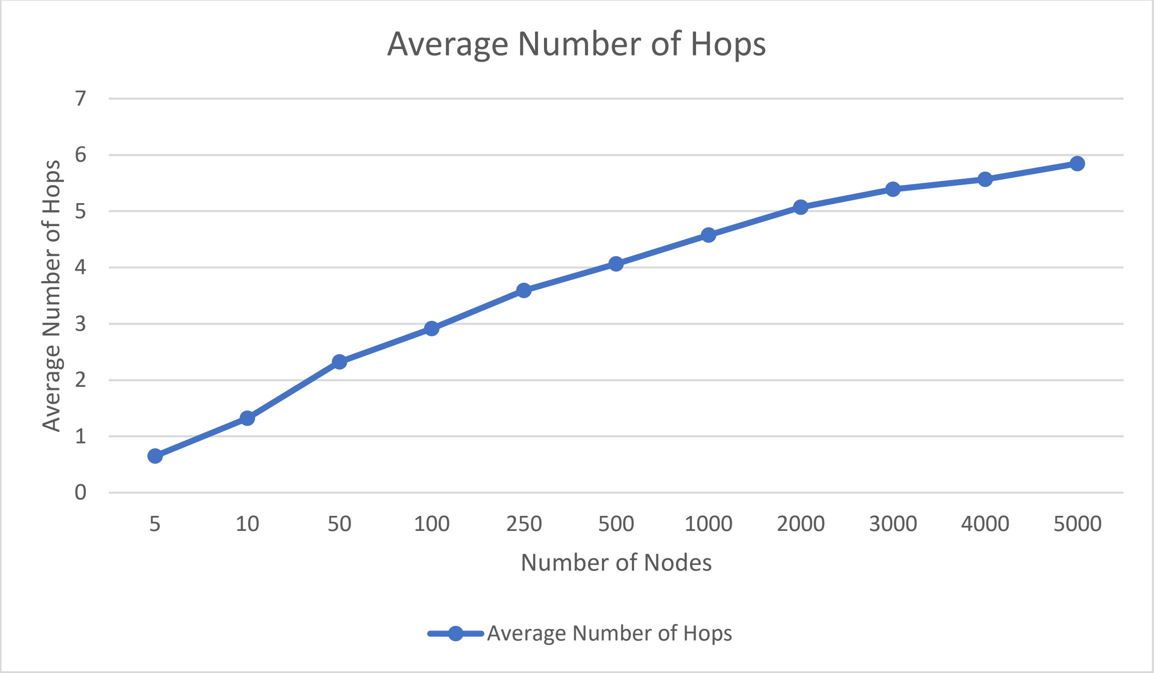
What is working

- Each node is added to the network consecutively and after every node has joined the network, the message starts getting transmitted to every other node.
- The transmission is terminated after each node gets requests from every node that sends them and subsequently transmits them.
- The average number of hops for the test cases ranged between 1.2 to 5.846. We can infer from this statistic that while searching in a node, the average number of hops would be about 3.
- Given, the number of requests to be delivered to every node using the chord protocol, the average number of hops (output) has been generated.
- All functionalities have been implemented and are working successfully.

Execution table

Number of Nodes	Number of Requests	Average Number of Hops	Total Number of Hops
5	5	0.960	24
5	20	0.650	65
10	3	1.200	36
10	10	1.310	131
10	20	1.320	260
50	3	2.260	339
50	10	2.322	1161
50	100	2.376	11883
100	10	2.915	2915
250	10	3.592	8981
500	10	4.064	20323
1000	10	4.572	457254
2000	10	5.068	101375
3000	10	5.391	161751
4000	10	5.566	222678
5000	10	5.846	292325

Plot - Number of Nodes v Average Number of Hops



Some Interesting Observations

- When the number of requests is greater than the number of nodes, we observed a slight increase (less than 1) in the average hope count.
- Larger the number of nodes in the network and smaller the number of requests, we observed a reasonable increase (greater than 1) in the average number of hops. The average hop values in this case were observed to lie between 2 and 5.
- When the number of requests and the number of nodes are equal (difference is negligible), the average number of hops are less than or equal to 1.
- Larger the number of nodes in the network, the program takes a substantial amount of time to finish execution.