COP5615: DOSP Project 3 Report
Chord - P2P System and Simulation

Name: Pranath Reddy Kumbam
UFID: 8512-0977

Problem Statement

Implement the Chord protocol and a simple object access service to prove its usefulness using Erlang and Actor Model based on the specification of the Chord protocol given by Stoica et al

language: Erlang

Implementation

The implementation follows the details mentioned in the problem statement and specifications detailed in the paper. For the hashing scheme, I have used SHA256, although, since it's a simulation and there is no sense of physical distance between the peers, the specific hashing scheme for the identifiers won't have an impact on the functioning and the output of the simulation as long as we choose an M that is sufficiently large enough to avoid collisions. The server first initiates a message and triggers the first node of the network, which then generates a random query and starts the lookup. Once the node has completed the required number of lookups, it passes on to the successor, and we continue execution until all the nodes have converged. Once all the nodes have been converged, we print the average number of communications/hops between the nodes.

Average No of Hops = (Total No of Hops)/(No of Nodes * No of Requests)

Execution Details

The program takes two parameters, number of nodes (numNodes), and the number of requests (numRequests).

Usage:
-> c(project3).
-> project3:main(numNodes, numRequests).
-> Ex: project3:main(1000, 10).
-> Restart shell after each execution

Example

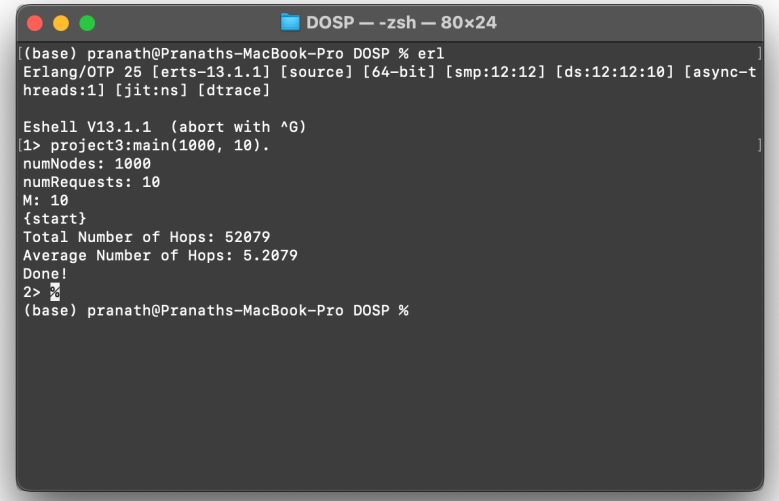

Figure 1: Example for 1000 nodes

Submission Details

The submission zip file contains the code, screenshots, results, and plots.

Results

This section presents all the results. As expected, we see a logarithmic relationship between the number of nodes and the average number of hops. Additionally, the average number of hops seems to be independent of the number of requests each node has to make. A table of all the results is submitted in the zip file.

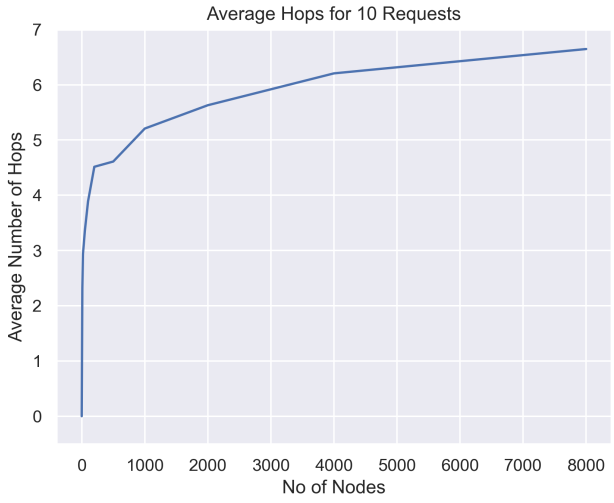

Figure 2: Plot for 10 Requests (Average Hops vs nodes)

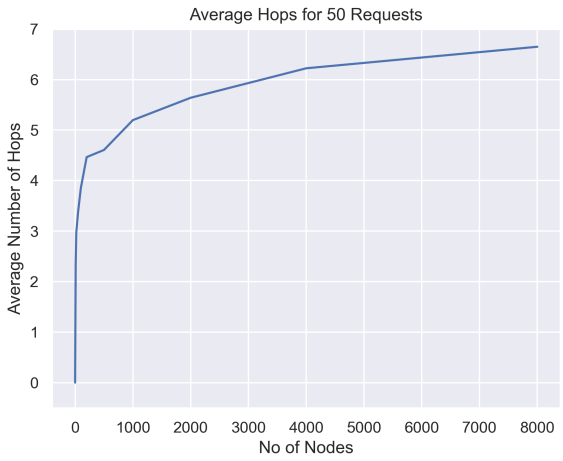

Figure 3: Plot for 50 Requests (Average Hops vs nodes)

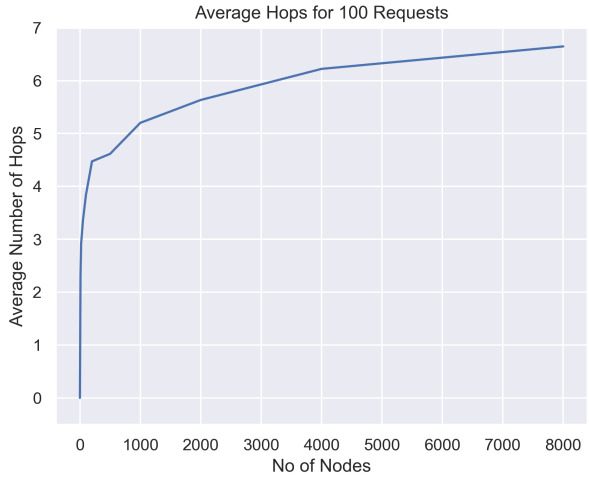

Figure 4: Plot for 100 Requests (Average Hops vs nodes)

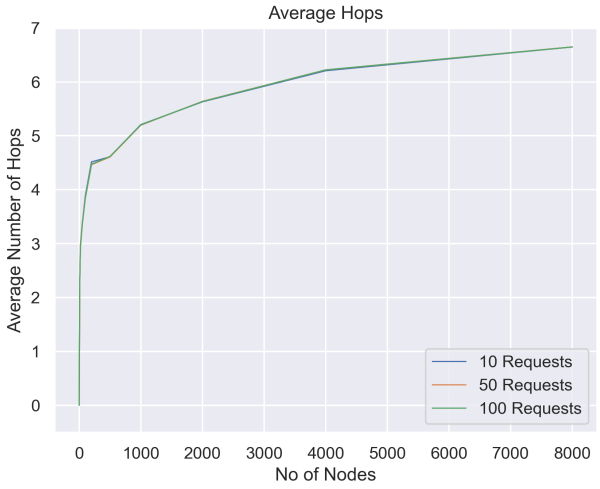

Figure 5: Comparison of Results (Average Hops vs nodes)

Log Plots

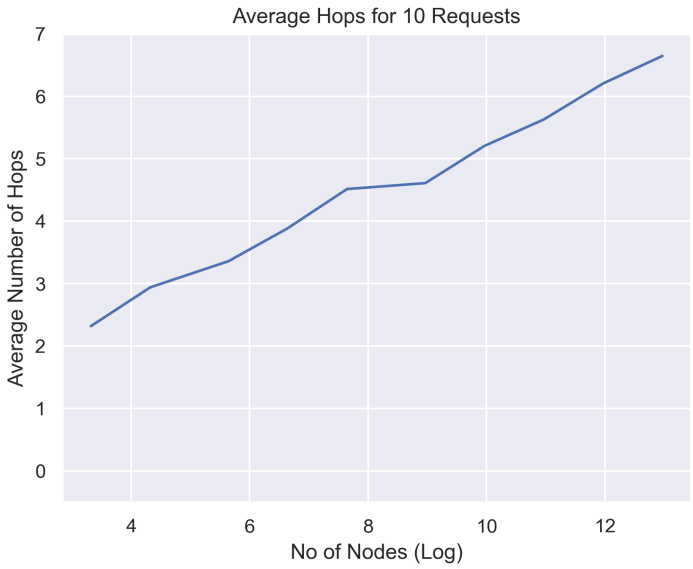

Figure 6: Log plot of Results for 10 Requests (Average Hops vs nodes)

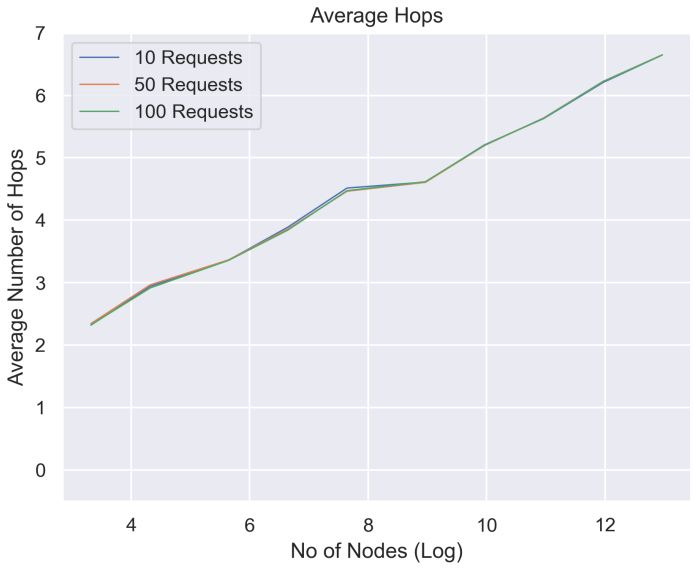

Figure 7: Comparison of Log Plots (Average Hops vs nodes)

Largest Network Run:

The maximum number of nodes tested is 10,000 for 10 requests and 50 requests per node.

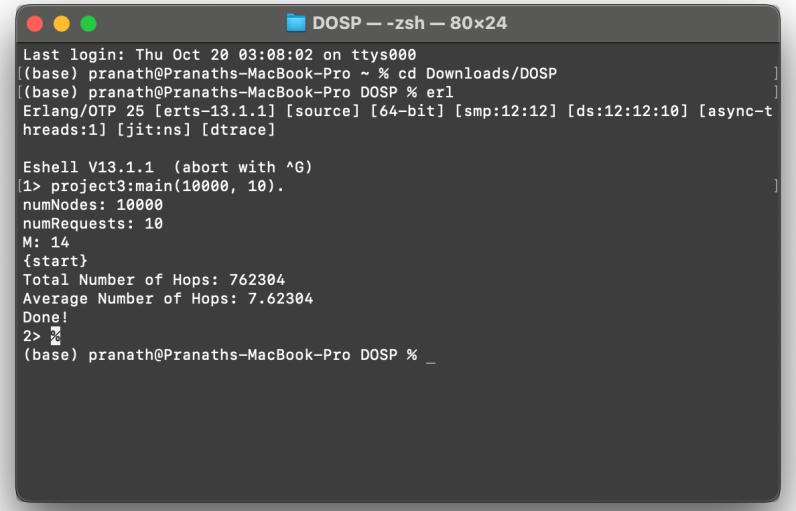

Figure 8: Output for 10,000 Nodes, 10 Requests

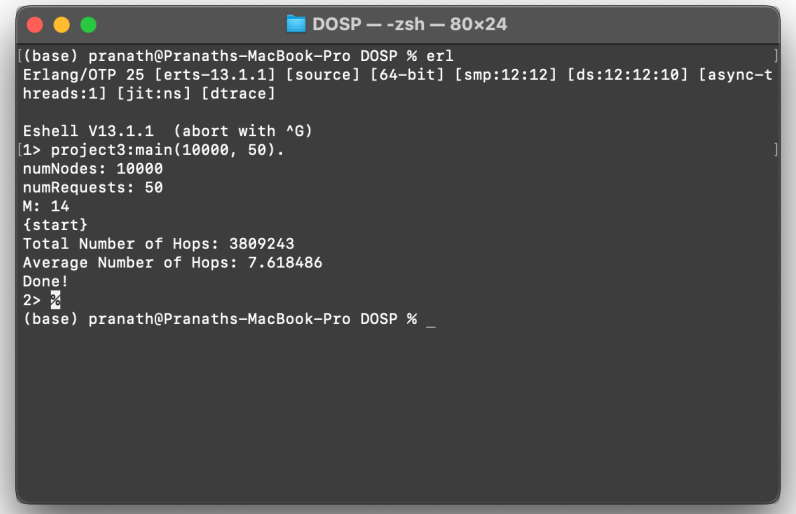

Figure 9: Output for 10,000 Nodes, 50 Requests