

Project 3 - Chord - P2P System and Simulation

COP5615 Distributed Operating Systems at the University of Florida.

Goal of the project

Using the Akka Actor framework, this project implements a simplified version of the Chord distributed hash table protocol. The system simulates Chord joins, where nodes can dynamically join the system, and lookups, where any node on the Chord ring network can be searched for a key.

With this implementation, logarithmic lookup performance is achieved as the number of nodes grows linearly, making use of the theoretically efficient way to construct the Chord ring.

Contribution

Ujwala Guttikonda, UFID 57914323, ujwalaguttikonda@ufl.edu

Nikhil Yerra, UFID 95453265, nikhil.yerra@ufl.edu

Steps to execute the code

Execute the following command

project3: start(numNodes, numRequests).

Input parameters :

- numNodes is an integer value which corresponds to the number of nodes you would like to input for the system.
- numRequests is an integer value which corresponds to the number of queries that each node must send and successfully respond to before shutting down. When all nodes are turned off, the system terminates.

What is working

The program is designed in such a way that it tracks the number of hops each request takes to be delivered and it will determine the average number of hops all the messages took to be delivered.

The maximum number of nodes which have been tested in the chord is 30000 nodes with 100 messages each. The average hop count for a message to arrive at its destination was in the range of [3, 5]. The result was 3.52 hops when the value was averaged over 9 runs. This implies that the chord protocol requires an average of 3.52 hops for every node to look for any content. It is evident from this that lookup requires time in the order of $\log n$.

Although more than 30000 nodes could be tested, stabilizing so many nodes would take a lot longer.

Largest network

In this project, the network has been tested for a maximum of 30000 nodes with 100 requests and the average number of hops is 7.101.

Results

Number of Nodes	Average number of hops
100	3.734
500	4.018
1000	4.498
5000	6.001
10000	6.499
15000	6.708
20000	6.999
25000	7.013
30000	7.101

Plot

