|  |  |
| --- | --- |
| Name | UFID |
| Aliya Abdullah | 5352-6307 |
| Raghav Gupta | 5897-9422 |

DOSP PROJECT - III

Chord - P2P System and Simulation

WHAT IS WORKING?

We have successfully implemented Chord Protocol as described in the research paper.

Our application accepts number of nodes that the chord structure will have, and how many times each node will try to find a key.

LARGEST NETWORK:

EXECUTING INSTRUCTIONS AND DETAILS:

Text in italics is meant to be run on command line.

To compile the code:

*c(project3).*

*c(node).*

To run the program:

*project3:start(50,5).*

Here, 10 is the number of nodes in the chord system and 5 is the number of requests.

The system will then print the average number of hops.

CREATING CHORD:

We take the approach of generating random keys as ID for each node instead of hashing, as the name space for SHA is too large to test on our personal computers.

We fix the name space using an arbitrary M value. The name space is .

Each actor is called and the value of PID is stored.

CREATING FINGER TABLES:

To create finger tables, we generate for 1≤i≤M as the Skey.

Each Skey is then mapped to corresponding nodes if the node is the smallest value larger than Skey.

Once a node has generated its finger table, it can be used for lookup operation. The listener is informed that all nodes have generated their respective lookup tables.

LOOKUP OPERATION AND CALCULATING AVERAGE NUMBER OF HOPS:

We generate random key values to lookup. Each node of the system looks up that value. Depending on the number of lookups given by the user, we generate that many random keys.

Each node reports how many hops it took to find the value. In the end, all these values are summed up and averaged.

If a system has NNode number of nodes, and user has entered NLookUp as the number of desired lookups, the average value is

Where Hops is the number of hops any one node took to find the key.