

COP5615 PROJECT 3 README

Group member Names:

Prapti Akolkar UFID: 22287019

Nikhil Kumar Singh UFID: 40500443

IMPLEMENTATION:

We performed the implementation of the Chord Protocol to solve the problem of P2P communication. It overcomes the difficulty of locating nodes and updating their locations when new nodes join the network or a particular node leaves it. There is no DNS-like system for peer-to-peer communication to resolve the location of the key node. To resolve a lookup operation in a maximum of $O(n)$ node calls, it saves minimal information about the other nodes in the network.

In order to implement the chord protocol for address resolution and resource storage, this project attempts to promote a peer-to-peer network.

To compile and run the code:

- Compiling the module using the `erl: c(chordprotocol)`.
- Starting the server: `chordprotocol:running_the_server(number of nodes, number of requests)`.
- The output is stored in `statistics.txt`

WHAT IS WORKING?

1. Nodes are successfully created and connected to the network. By making calls to other nodes in the network, each node creates its own finger table.
2. By performing at most $O(\log(N))$ recursive queries to other nodes, look-up functionality can locate the key.
3. The average number of hops shown at the conclusion is always lower than the $\log(N)$ and fits our expected outcome.
4. Chord Protocol simulation has been executed successfully meeting all the requirements of this assignment.
5. There is a finger table that is linked with every node and has the value related to node numbers.
6. We considered for this project a fixed number of messages i.e., 10

WHAT IS THE LARGEST NETWORK YOU MANAGED TO DEAL WITH:

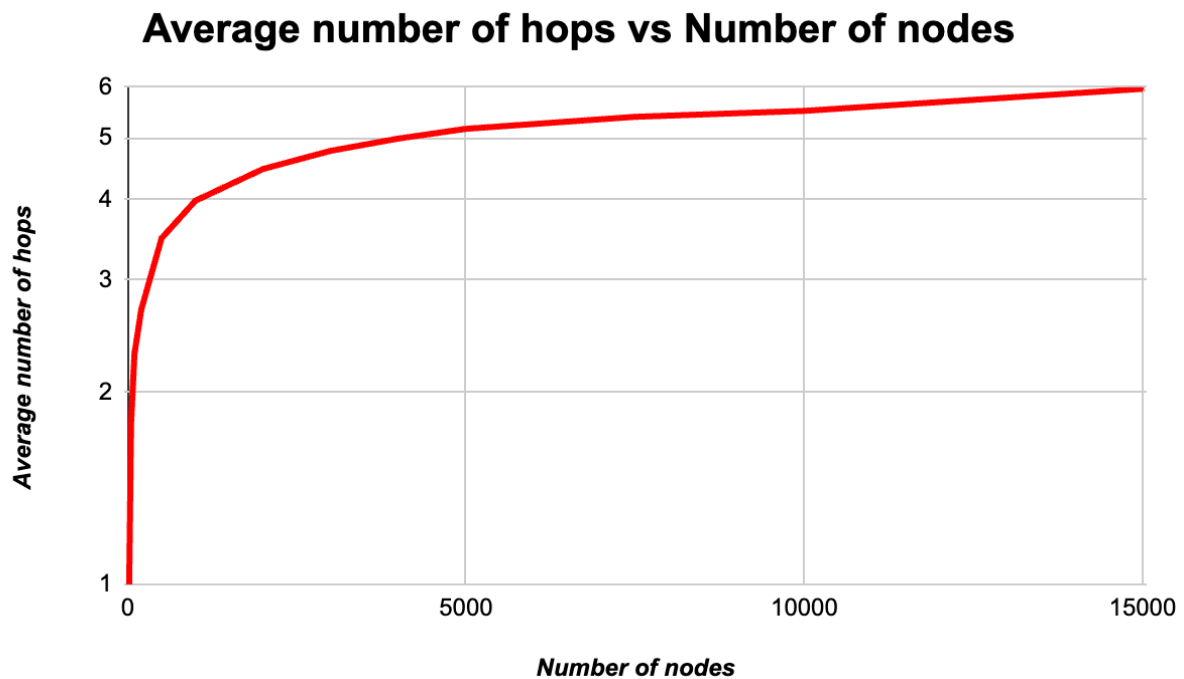
The count of the average hops is the total hops count per total number of requests.

The maximum number of nodes we handled is 15000 nodes with fixed number of messages=10 with the maximum average number of hops being 5.96706 and total number of hops=895059

GRAPHICAL REPRESENTATION :

X-axis: Number of nodes

Y-axis: Average number of hops



Data:

Number of nodes	Average number of hops
10	0.79
50	1.788
100	2.295
200	2.691
500	3.4814
1000	3.9863

2000	4.46955
3000	4.772
4000	4.9875
5000	5.166
7500	5.39405
10000	5.5109
15000	5.96706

OUTPUTS:

The results of the project are illustrated below:

```

● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(50,10).
true

Average Number of Hops = 1.788   Total Number of Hops = 894   Number Of Nodes = 50   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(100,10).
true

Average Number of Hops = 2.295   Total Number of Hops = 2295   Number Of Nodes = 100   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(200,10).
true

Average Number of Hops = 2.691   Total Number of Hops = 5382   Number Of Nodes = 200   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
○ nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(500,10).
true

Average Number of Hops = 3.4814   Total Number of Hops = 17407   Number Of Nodes = 500   Number Of Requests = 10
3>

```

```

● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(1000,10).
true

Average Number of Hops = 3.9863   Total Number of Hops = 39863   Number Of Nodes = 1000   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(2000,10).
true

Average Number of Hops = 4.46955   Total Number of Hops = 89391   Number Of Nodes = 2000   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(3000,10).
true

Average Number of Hops = 4.7722333333333   Total Number of Hops = 143167   Number Of Nodes = 3000   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
○ nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(4000,10).
true

Average Number of Hops = 4.9875   Total Number of Hops = 199500   Number Of Nodes = 4000   Number Of Requests = 10
3>

```

```

● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(5000,10).
true

Average Number of Hops = 5.16606   Total Number of Hops = 258303   Number Of Nodes = 5000   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(7500,10).
true

Average Number of Hops = 5.3940533333333   Total Number of Hops = 404554   Number Of Nodes = 7500   Number Of Requests = 10
3>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
● nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(10000,10).
true

Average Number of Hops = 5.5109   Total Number of Hops = 551090   Number Of Nodes = 10000   Number Of Requests = 10
3>

```

```
nikhilkumarsingh@Nikhils-MacBook-Air Project3 % erl
Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.0.4 (abort with ^G)
1> c(chordprotocol).
{ok,chordprotocol}
2> chordprotocol:running_the_server(15000,10).
true

Average Number of Hops = 5.96706   Total Number of Hops = 895059   Number Of Nodes = 15000   Number Of Requests = 10
```

OBSERVATIONS & CONCLUSION:

We observed from the graph representing number of hops vs number of nodes that average number of hops increases in logarithmic fashion against number of nodes. Further we noticed that average number of hops is independent of the number of messages sent and only depends on the number of nodes. As we increase the number of nodes and number of messages the total number of combinations of interaction between the nodes has increased significantly leading to large processing time. Thus, we implemented the Chord protocol as per the project requirements.