Project 3

Group Members

Raj Vora - 35551411

Rushil Patel - 66999320

Problem Definition

The goal of this project is to implement the Chord protocol and a simple object access service to prove it's usefulness using Erlang and Actor Model.

Compile

```
> erl
> c(chord).
> c(peer).
```

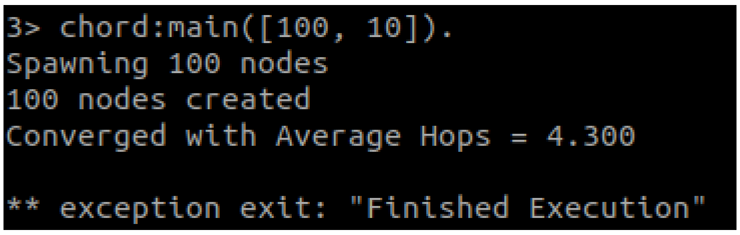
Execute (erlang shell)

```
> erl
> chord:main([Nodes, Requests]).
```

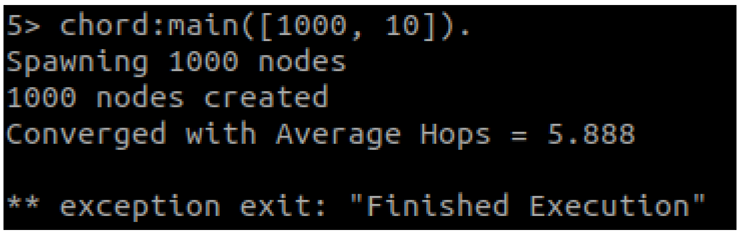
Where `Nodes` is the number of peers to be created in the peer-to-peer system and `Requests` is the number of requests each peer has to make. When all the peers complete that many requests, the program will exit. Each peer sends a request/second.

Working

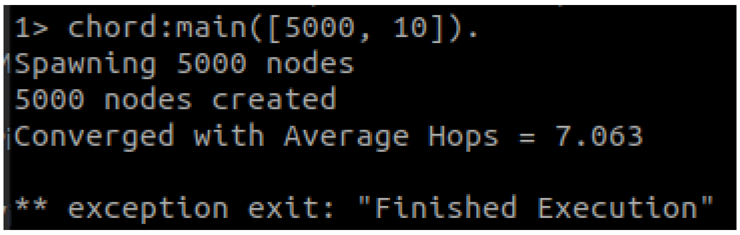
We managed to run Chord protocol for maximum 10000 nodes and 100 messages



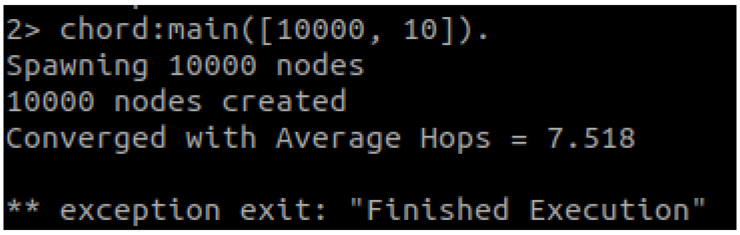
```
3> chord:main([100, 10]).
Spawning 100 nodes
100 nodes created
Converged with Average Hops = 4.300

** exception exit: "Finished Execution"
```



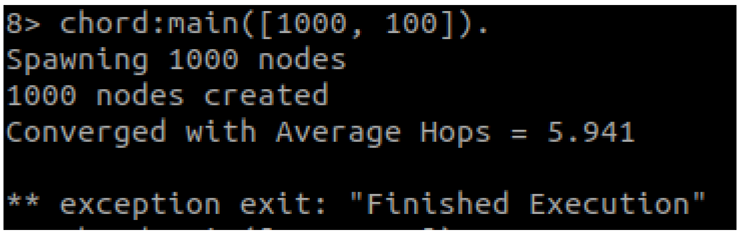
```
5> chord:main([1000, 10]).
Spawning 1000 nodes
1000 nodes created
Converged with Average Hops = 5.888

** exception exit: "Finished Execution"
```



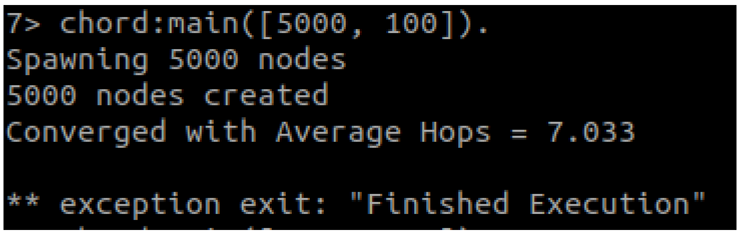
```
1> chord:main([5000, 10]).
Spawning 5000 nodes
5000 nodes created
Converged with Average Hops = 7.063

** exception exit: "Finished Execution"
```



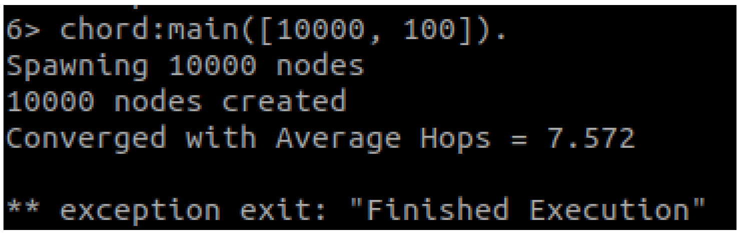
```
2> chord:main([10000, 10]).
Spawning 10000 nodes
10000 nodes created
Converged with Average Hops = 7.518

** exception exit: "Finished Execution"
```



```
8> chord:main([1000, 100]).
Spawning 1000 nodes
1000 nodes created
Converged with Average Hops = 5.941

** exception exit: "Finished Execution"
```



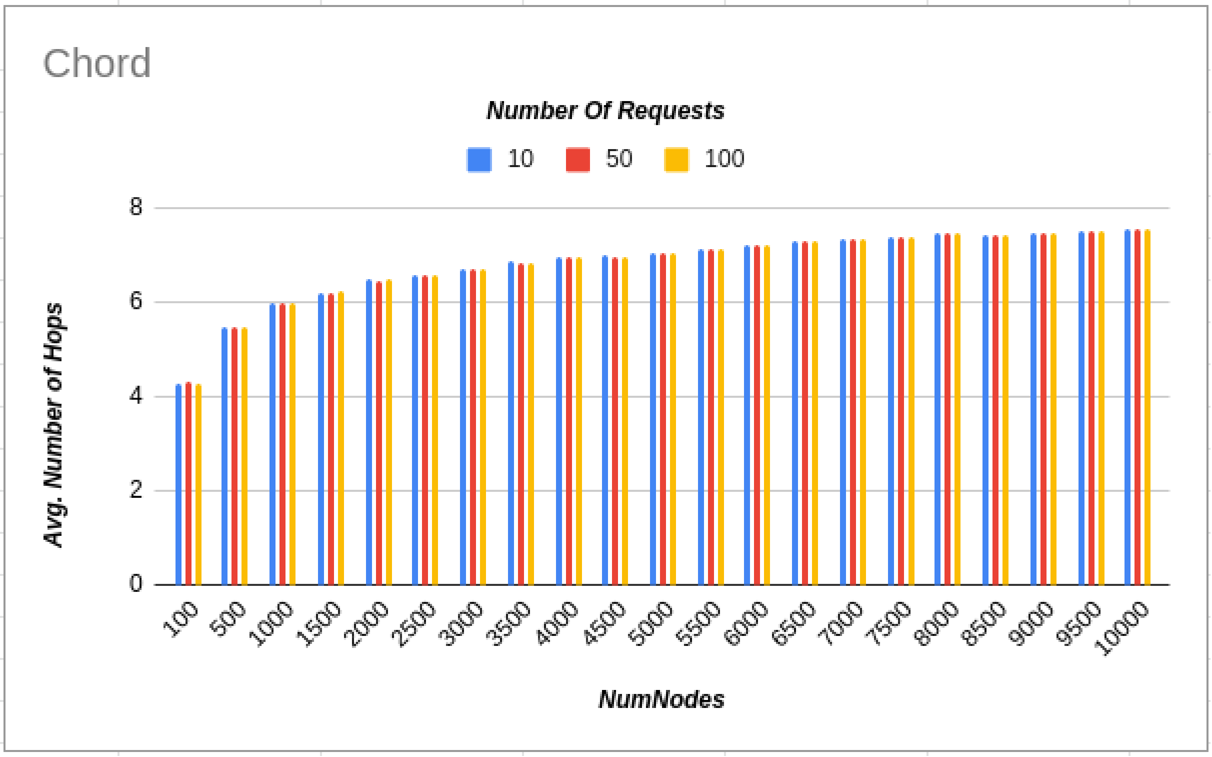
```
7> chord:main([5000, 100]).
Spawning 5000 nodes
5000 nodes created
Converged with Average Hops = 7.033

** exception exit: "Finished Execution"
```



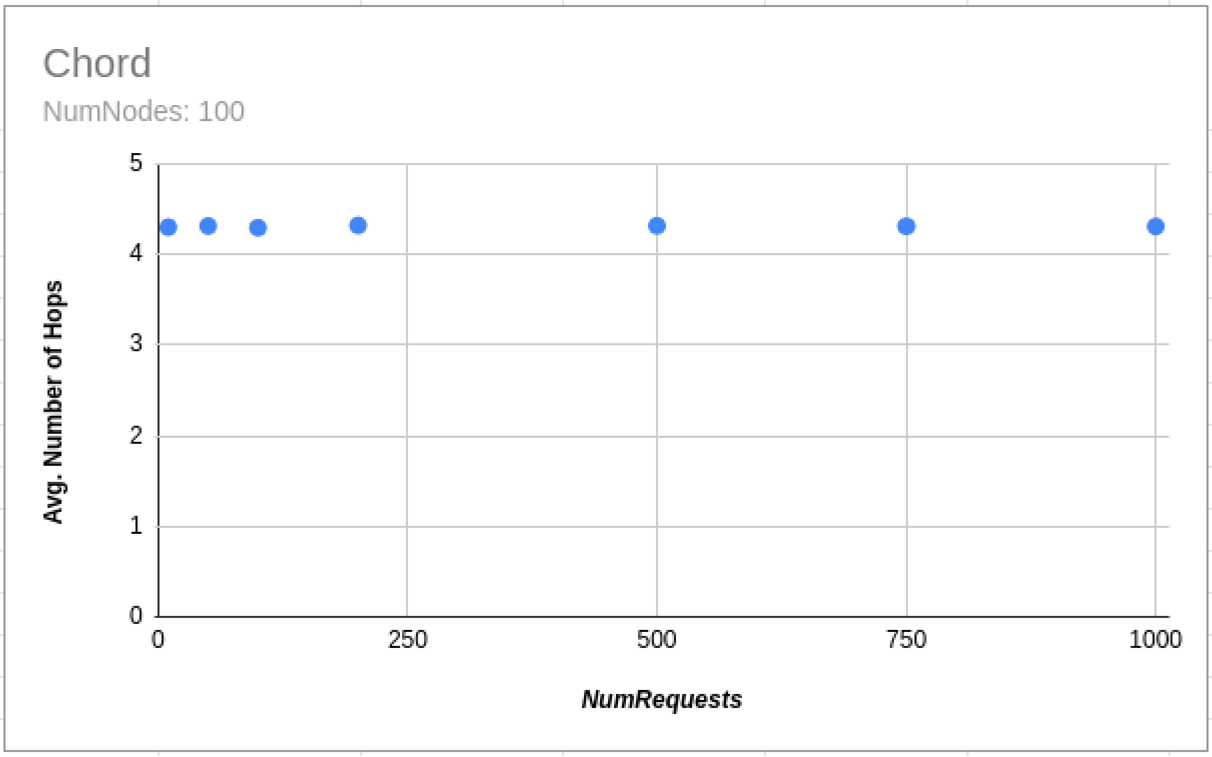
```
6> chord:main([10000, 100]).
Spawning 10000 nodes
10000 nodes created
Converged with Average Hops = 7.572

** exception exit: "Finished Execution"
```

Some Observations

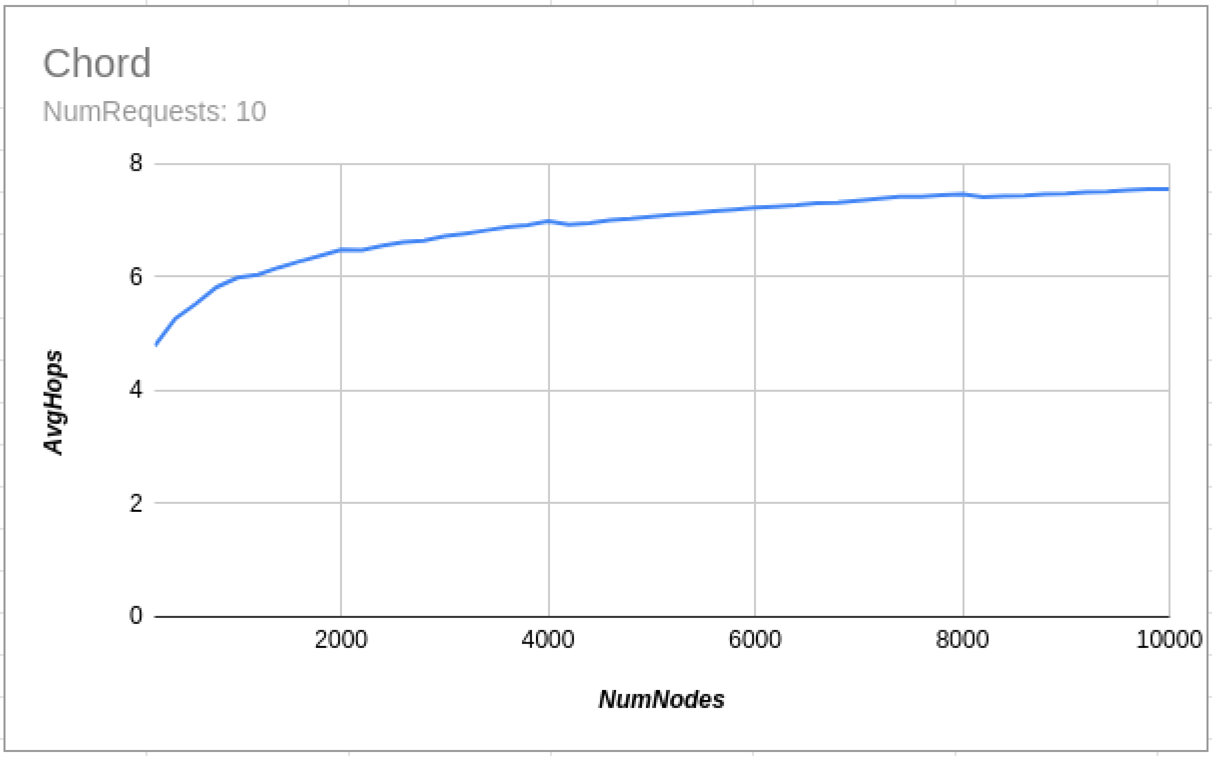
- Average Hops primarily depends on the Number of Nodes doesn't really depend on the Number of Requests.



- Number of Requests mainly elongates the running time of the program and adds some random noise to the output and nothing else.



- As the number of nodes increases, average hops also increase but in a logarithmic fashion


Chord
NumRequests: 10
AvgHops
NumNodes
0
2
4
6
8
2000
4000
6000
8000
10000