Project 3 (Chord P2P System)

Group Members:	UF ID:
Kalpana Sathya Ponnada	5246-1920
Tharun Bhupathi	8328-9089

What is Working:

- All the required methods which are mentioned in the chord paper are implemented as per their functionality.
- For the total number of resources as given by the user, the number of hops taken to find each resource is found and hence, the total number of hops for all the resources is calculated.
- The average hops taken are calculated by dividing the total number of hops by the total number of resources or keys.

Implementation Details:

- The Chord peer to peer system is implemented as per the details mentioned in the section IV of the paper.
- Initially, we spawned the total nodes as per the user input sequentially and at last a supervisor node to monitor the nodes
- When the first node joins the network, it doesn't have any predecessor and successor. A create function is called to add this node to the network. We also register this node as the global node for future reference.
- Later, when each node joins, its various parameters are like predecessor, successor, finger table (consisting of Logarithmic number of entries to store the various nodes info) etc. are updated with the help of modules.
- At every stage, when a node joins the network, we call the find_successor of this new node on the global node and it returns the successor. This is updated in the new node's successor parameter and the finger table first index. At this stage, the successor is not notified about the presence of new node.
- In periodic Intervals, various methods are called whose duty is to update the successor of each node and notify the successor about its predecessor, update the finger table of every node, checks whether the predecessor is alive or not respectively.
- Once all the nodes are added to the chord ring and are stabilized, we find the number of hops taken by node to get the required key. This is done for all the number of required keys (Which is taken from user as input). Finally, average number of hopes taken is calculated as $\frac{Total\ Number\ of\ Hops}{Total\ Number\ of\ Requests}$
- With the implemented chord algorithm, the maximum number of hops a node takes to find the required key is $\log_2 N$, where N is the total number of nodes in the network.

Input:

Number of Nodes, Number of Requests

Steps to compile:

- Compile the following files in erlang shell

project3.erl

```
1> c(project3).
{ok,project3}
```

Steps to Run:

- project3:start()

```
2> project3:start().
```

Parameters:

We take following as the command line input from the user:

1) Number of Nodes
2) Number of Requests

```
Enter the number of nodes: 10
Enter the number of requests: 10
true
```

Output:

- Mean of the hops for the total number of peers along with the logarithm of number of peers

a) Number of peers=20

```
5> project3:start().
Enter the number of nodes: 20
Enter the number of requests: 30
true
6>
 Average Number of Hops = 1.1033333333333333  r6>
 Total Number of Nodes = 20  6>
 Logarithm of number of nodes to the base2 = 4.321928094887363  6>
```

b) Number of peers=50

```
1> project3:start().
Enter the number of nodes: 50
Enter the number of requests: 55
true
2>
 Average Number of Hops = 1.7945454545454544  2>
 Total Number of Nodes = 50  2>
 Logarithm of number of nodes to the base2 = 5.643856189774724
```

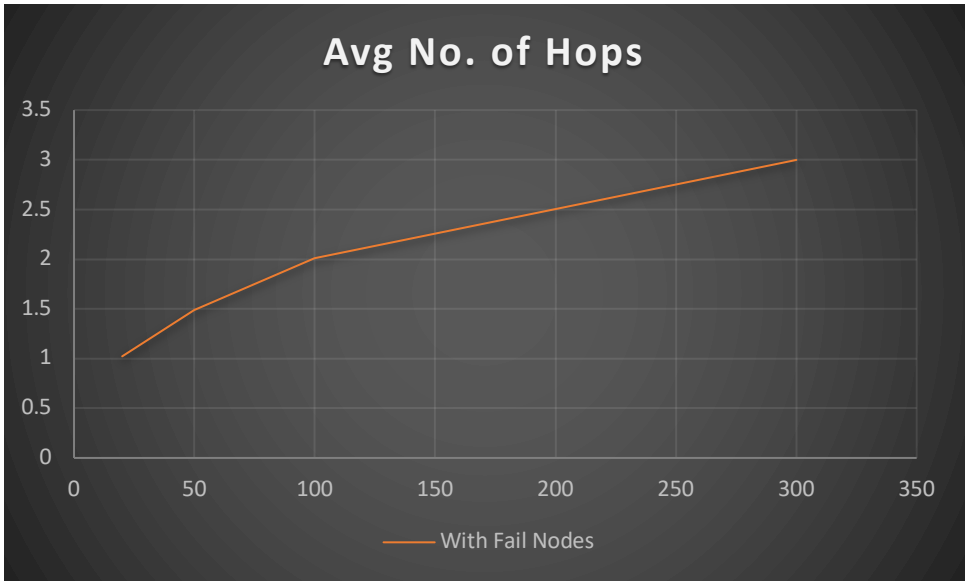
c) Number of peers=100

```
1> project3:start().
Enter the number of nodes: 100
Enter the number of requests: 150
true
2>
 Average Number of Hops = 2.271733333333333  2>
 Total Number of Nodes = 100  2>
 Logarithm of number of nodes to the base2 = 6.643856189774724
```

d) Number of peers=300

```
1> project3:start().
Enter the number of nodes: 300
Enter the number of requests: 400
true
2>
 Average Number of Hops = 3.0421583333333335  r2>
 Total Number of Nodes = 300  2>
 Logarithm of number of nodes to the base2 = 8.228818690495881
```

Variation of Avg Number of Hops with the Number of nodes



System Configuration:

MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)
2.3 GHz Dual-Core Intel Core i5 processor

The largest network we managed to find:

- The network with 5,000 nodes is the largest one we were able to identify.

```
Average Number of Hops = 5.19624
Total Number of Nodes = 5000
Logarithm of number of nodes to the base2 = 12.287712379549449
```