PROJECT 3 – PASTRY

Sakshi Garg
9750-4936
sakshigarg.1@ufl.edu

The goal of the project is to implement the pastry protocol using the actor model in elixir.

Code Submitted:

project3.ex
worker.ex

How To Run:

```
mix escript.build
./project3 numNodes numRequests
```

What is working:

Each node has a unique identifier which is called nodeId. When the node is provided with the message and the key, then that Pastry node routes the message to the node which has the nodeId closest to the key. It will check for the node amongst all the live Pastry nodes in the network. Each pastry node keeps track of its immediate neighbours in the nodeId space and even notifies applications on the arrival of any new node.It also manage network locality and minimises the distance messages travel, according to the number of IP routing hops.

We tested the complete functionality of Join and Routing aspects of Pastry Algorithm.
The output will be the number of log measures indicating the events occurring. The final and end line contains the average number of hops that were required to be calculated.

- It takes the input as number of nodes(numNodes) to be created in the network and number of requests(numRequests) that each node has to make.
- Each Pastry node is being initialised and maintains the routing table and leaf nodes. It is responsible for the message routing to the appropriate peer.
- Once the network is built it sends number of messages to each of the pastry node actor specified by number of requests.
- If the message is received at the destination node then it maintains the number of hops per node and at the end when all messages are received at the destination average number of hops are calculated which is then printed.

What is the largest network we can manage:

Larger network may be supported, but may take a longer time.
largest numNodes -> 70 and numRequests -> 5

numNodes	numRequests	average no of hops
10	2	1.875
15	5	1.566
20	5	1.8125
20	6	1.69
20	7	1.97
20	8	2.08
20	9	1.79
20	10	1.89
30	2	2.0
30	5	1.95
40	2	2.0625
50	2	2.25
50	5	2.13125
50	8	3.39
60	2	2.27
70	2	2.34
70	5	3.89
80	2	2.46
100	2	2.71