

README

TEAM MEMBERS

Individual Project

RUNNING THE CODE

Run project2.fsx

```
dotnet fsi --langversion:preview .\project3.fsx numNodes numRequests
```

WHAT IS WORKING

- Everything :)
- I have implemented the pastry protocol as described in the paper
- I am able to replicate the results expected by the paper i.e. the average number of hops per request is $\log_{16} N$
N = number of nodes
 $16 = 2^b = 2^4$

IMPLEMENTATION

1. The id for each node is chosen at random from the range [1, 10000000] and converted to hexadecimal. So, $b=4$ (as defined in the paper)
2. Join
 - a. The nodes are added one by one to the network
 - b. A new node joins by contacting one of the already existing nodes and sending it "Join" message
 - c. The node that was contacted then routes the "Join" message in the network
 - d. The nodes that are visited while routing give the corresponding row from their routing table and leaf set (if necessary) to the newly joined node
 - e. The new node updates its routing table and leaf set accordingly
3. Route
 - a. When a Route message is received, a node first check the destination is itself in which case it consumes the message
 - b. If the destination is closest to this node than all the nodes in its leaf set then it consumes the message
 - c. If the destination is in the range of the leaf set then it forwards the message to the node that is closest to the destination
 - d. If the destination is not in the range of the leaf set then it uses its routing table. It tries to match the longest prefix it can (i.e. tries to route the packet as far as possible) and then forwards based on the bit that doesn't match
 - e. If there is no entry in the routing table then it checks in the previous row

LARGEST PROBLEM

Average number of hops for different problems

	100 nodes	1000 nodes	10000 nodes
1 request/node	0.95	1.994	3.3417
10 requests/node	0.92	2.0329	3.57686
50 requests/node	0.95	2.24	3.966448
100 requests/node	0.96	2.113	3.60659

Although I haven't tried larger number of nodes than this (because the Join phase takes up a lot of time) but I am sure that all the requests will be served in $\log_{16} N$ hops where N =number of nodes

RESULT

```
dotnet fsi --langversion:preview .\project3.fsx 10000 100
```

```
PS D:\Workspace\F#\Distributed-Operating-System\Project 3> dotnet fsi --langversion:preview .\project3.fsx 10000 100
Join phase: Please wait for all the nodes to join the network
All nodes have joined and routing tables have converged
Request phase: Nodes will now make 1 request per second
All requests have been served
Average hops per requests = 3.793742
PS D:\Workspace\F#\Distributed-Operating-System\Project 3> |
```

CONCLUSION

We used Actor model to implement Pastry protocol without having to worry about threads and mutexes