Fall 2020

COP5615 Distributed Operating System Principles

Readme file

Project 3

Group Details

Parth P. Chitroda	Saheel R. Sawant
5189-1737	1164-7923
pchitroda@ufl.edu	sawant.s@ufl.edu

- **What is working?**

In this project, we have implemented the pastry protocol using fsharp's actor model. In pastry protocol, each peer needs to maintain a routing table and a leaf set. For any peer, the leaf set contains a set of nearest neighbor nodes where the message could be sent. A routing table has $\log_{2^b} N$ rows where N is the total number of peers. For b = 4, the routing table can have ($\log_{16} N$) rows.

Upon receiving the message, each peer first checks whether the key lies in its leaf set. If the key is present, then the message is sent to the peer whose nodeId is closest to the key. If the key is absent, then the routing table of the peer is accessed and the message is sent to the peer with the closest common prefix.

Each node must perform the said number of the requests, before terminating and displaying the average number of nodes or hops required to successfully deliver the message. The following table shows the average number of hops obtained for varying numNodes.

How to run the project :
dotnet fsi --langversion:preview project3.fsx numNodes numRequest

Two user inputs are taken for this implementation. First, numNodes which is the total nodes required to form a peer to peer network and numRequests which is the number of requests each node in the network has to make.

numNodes	numRequests	Avg. Number of Hops
10	10	1.35
50	10	2.05
100	10	2.19
500	10	2.86
1000	10	3.10
5000	10	3.57
10000	10	3.92

- **What is the largest network you managed to deal with?**

The largest network we were able to solve has numNodes equal to 10000 for numRequests equal to 10.

Bonus:

In this part, we have implemented a failure model for the above implementation. Whenever a number of nodes in a network fails, there are two parts of the network that need to be fixed. Firstly, the dead node IDs that exist in the leafset of the other peers and the dead node IDs that exist in the routing table of other nodes.

While updating a routing table, first the peers which have idle IDs are removed. After that, a message is sent to the peers that are closer to the idle peer. Based on their response, the routing tables are further updated. Also, while updating the leafsets, the peers in every dead node's leafset are removed and then to refill the leaf set a message is sent to the last node which lies near the idle node.

How to run :

```
dotnet fsi --langversion:preview project3-bonus.fsx numNodes numRequest numFailures
```

The average number of hops will first increase because some of the nodes will not get an accurate replacement of the idle node's ID. But, the average hops decreases as the number of nodes in the network significantly reduces.

numNodes	numRequests	numFailures	Avg. Hops
100	10	25	2.04
100	10	50	2.21
100	10	75	2.45
100	10	90	2.25

The average number of hops decreases as the number of failure nodes in the network increases, due to fewer number of nodes available in the network. But, even with lesser peers, the message does reach the destination node.

numNodes	numRequests	numFailures	Avg. Hops
100	10	20	2.15
500	10	100	2.78
1000	10	200	3.10
5000	10	1000	3.42
100	10	40	1.99
500	10	200	2.37
1000	10	400	2.87
5000	10	2000	3.35