Project 3 Pastry

1. Team Members

Name: Zixun Wang UFID: 3725-9823
Name: Yixin Wei UFID: 5114-6181

2. Observations

We implemented the Pastry network with b = 4, and L = 16. For the purpose of routing, nodeIds and keys are though of as a sequence of digits with base 16 (2^b).

We assume a network consisting of numNodes nodes. And Pastry can route to the numerically closest node to a given key in less than $\log_{16}$(numNodes) steps under normal operation.

Given a message, the node first checks to see if the key falls within the range of nodeIds covered by its leaf set, if not, the node will check the routing table. And the routing table has approximately $\log_2{}^b$(numNodes) rows with 2^b-1 entries each.

If we create some fail nodes in the program, when we meet a node which nodeId equals to the nodeId of fail nodes, we would choose another node. If any nodes on the path are failed, then the source will find a closest node and route the information to that node. And we need to update the dead nodes lying in the leaf set and the routing table. We need to delete dead nodes in them and update the leaf set and routing table.

Because some nodes will have to replace their nodeId with nodes not failed, the average number of hops will increase. And also, the average hops will decrease a bit due to dead nodes lead to fewer nodes in the Patry network.

Here is a table shows the change of Pastry work when some nodes failed.
When numRequests = 10:

numNodes	fail nodes	average hops
100	0	1.4
	10	1.44
	20	1.4875
	50	1.46
500	0	1.882
	50	1.926
	100	1.9125
	250	2.01

As shown in the table, the overall average hops will increase a bit, and we observed that Pastry network is able to self-organize and adapt to node failure.

A link to a screen capture of the running program is: https://recordit.co/0Qw4VWRMAL