PROJECT-3
PASTRY

Lavanya Chennupati(UFID: 1129-9429)
Sowmya Duggimpudi(UFID: 1171-2252)

Working

Pastry is self-organizing routing overlay protocol, where each node routes the message until the message reaches the destination.

Implementation

Pastry is implemented mainly in the overlay networks in which each node is assigned a unique identifier(nodeId). The network is created based on the number of the nodes given through the input value numofNodes. The id assigned to a node is 128 bit long and is assigned to its physically closest node based on the proximity metric.

When a single node successfully joins the network, the subsequent nodes are started by the pastry manager. When a new node joins the network, its leafset, routing table and the neighborhood set are updated based on the node which it already knows in the network. After the updation is done, its state is sent to all the nodes in its table so that their entries can be updated. Once it is done the new node sends an acknowledgement to the Manager which approves that the node can join the network. The nodes can be changed dynamically in the network i.e. a node can join or leave the network as needed,

Now the nodes can send the messages to one another and the number of requests is given by the variable numofRequests. Once the messages are sent successfully, the Manager is notified which calculates the average number of hops. This converges to the value log(numofNodes)/log16. We take log16 as the base of the node id is 16 (b=4).

Steps to Run

Compile : scalac project3.scala

Run : scala project3 <numofNodes><numofRequests>

Example : scala project3 500 20

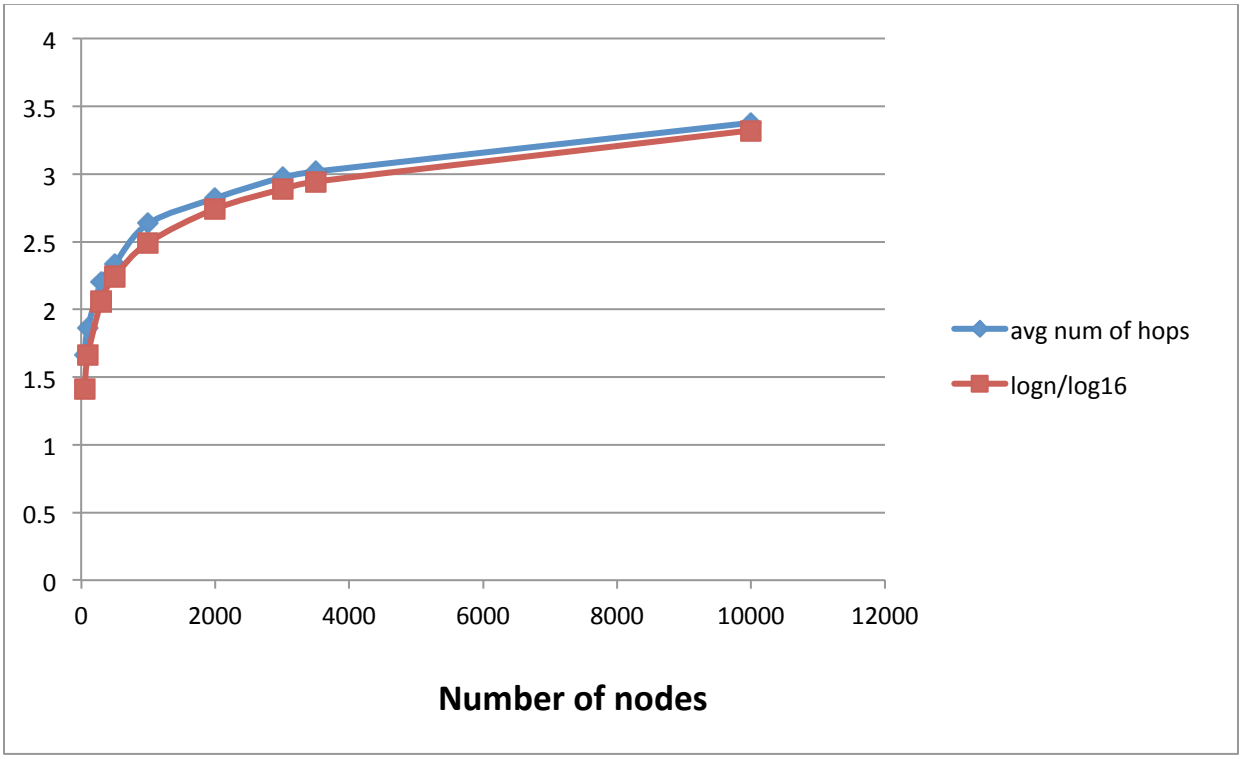
Largest Problem Solved

Observations:

Number of Nodes	Number of Requests	Average No. of Hops	Log(numofNodes)/Log(16)
50	15	1.664	1.41
100	40	1.862	1.66
300	120	2.205	2.057
500	200	2.336	2.242
1000	300	2.633	2.492
2000	500	2.821	2.742
3000	1000	2.974	2.888
3500	1200	3.018	2.944
10000	100	3.378	3.322

The largest network we managed to deal with is a network of 10000 nodes and 100 requests

GRAPH :


4
3.5
3
2.5
2
1.5
1
0.5
0
0
2000
4000
6000
8000
10000
12000
Number of nodes
avg num of hops
logn/log16