ADVANCED DISTRIBUTED SYSTEMS (COL819)

2020MCS2444: Aayush Kumar Singh

March 2021

IMPLEMENTING PASTRY

1 INTRODUCTION

Pastry is a scalable, distributed, peer-to-peer, overlay network of nodes for sharing files and data. This project deals with the implementaion and evaluation of the system under many circumstances.

2 DESIGN DETAILS:

The design follows directly from "Pastry:Scalable, decentralized object location and routing for large-scale peer-to-peer systems" paper published by Anthony Rowstron and Peter Druschel.

Pastry Node:
Initially, we assume each node has an IP address (32-bit) and a location coordinate specified by in a plane of with center of plane at (0,0) with x ranging from -180 to +190 and y ranging from -90 to +90, similar to GPS coordinates. These parameters are unique for all the nodes that are present in system. Each node can store some data and has a key associated with each data item which is numerically closer to node ID of the node.

These nodes in system will only be used to initialize the pastry network. The process of initialization we take a node and it's IP address using MD5 hashing algorithm generate a 32 bit Node ID which is unique to each node. It initializes it's routing table and leaf set with it's own Node ID at appropriate locations. Then, using the geographical nearest node it finds the node ID of node whose node ID is numerically closest to Node ID of joining node.

During this process, it stores all the nodes encountered in the routing path and fill this node's leaf set and corresponding rows of it's routing table. After

this node joins the Pastry network it let's all other node know his presence by sending it's own ID to entries in it's leaf set, neighbouring set and routing table.

Pastry Network:
The pastry network keeps track of: i. All the nodes which have an IP address and a location.
ii. Nodes which have joined the network.
iii. Nodes that were safely deleted.
The deleted nodes are kept track of in order for them to join again if they want.

2.1 Operations:

i. Insert a node. A node in our network can be inserted either from the set of deleted nodes or by explicitly proving an IP address and location.

ii. Delete a node. A node is safely deleted from network by informing the nodes in it's leaf, neighbourhood and routing table of it's departure so they can fix their corresponding entries.

The data stored in this node will be sent to neighbouring node in leaf set or will be re-distributed accordingly if no nodes present in leaf set. This ensures data is not lost.

iii. Look up data. We can look up data stored in one of nodes present in network.

iv. Add key-value pairs For adding data we first convert the the data to a 128 bit hash using MD5 and then find the node closest to this node ID and store the data in it.

v. Print Routing Table (With summary) The routing table of a node along with summary of network: total no of nodes, total number of data elements etc can be printed given a Node ID.

3 EVALAUATION OF SYSTEM:

3.1 Configurations of Network:

The value chosen for parameters:
i. b = 4
ii. L = 10
iii. M = 10
iv. Routing Table size: 32 x 16

3.2 Evaluation: Part I

In this part we evaluate over system with above parameters and varying the no of nodes in network from 100 to 500 to 1000.

Populating the network:
In each configuration, we first randomly pick a three letter string and a random node that is present in network using which we store the string in appropriate node in network. This is repeated for 10,000 times.

Querying data:
In each configuration, we select a random node and query one of the strings stored while populating. This is repeated 1 million times.

No of nodes in network: 100

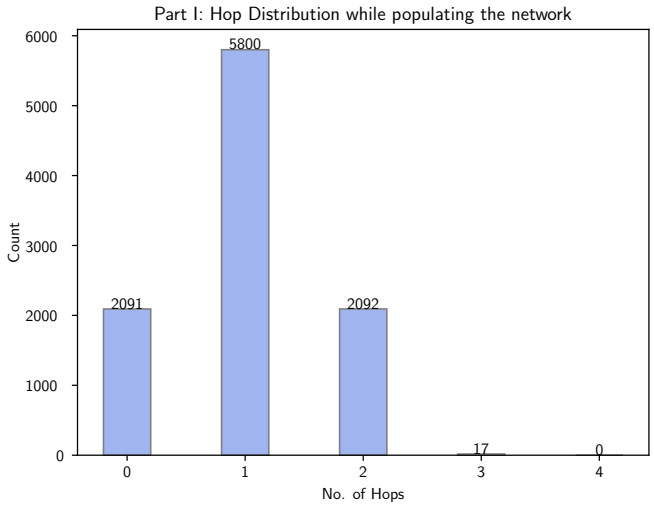

Figure 1: Hop Distribution while populating the network when no of nodes in network is 100.

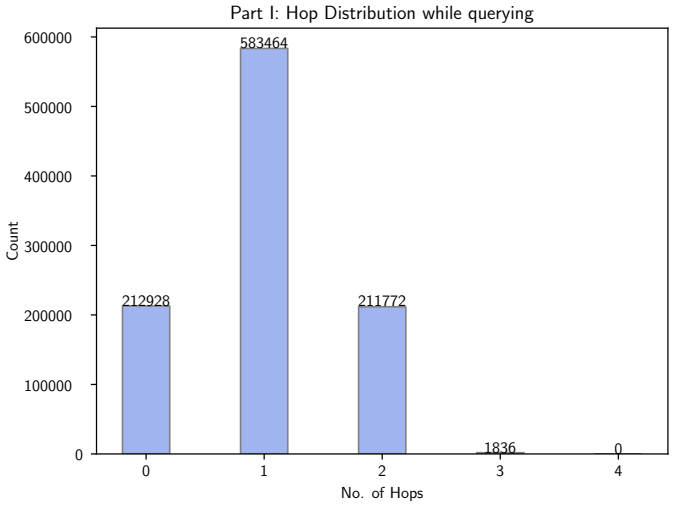

Figure 2: Hop Distribution while querying when no of nodes in network is 100.

No of nodes in network: 500

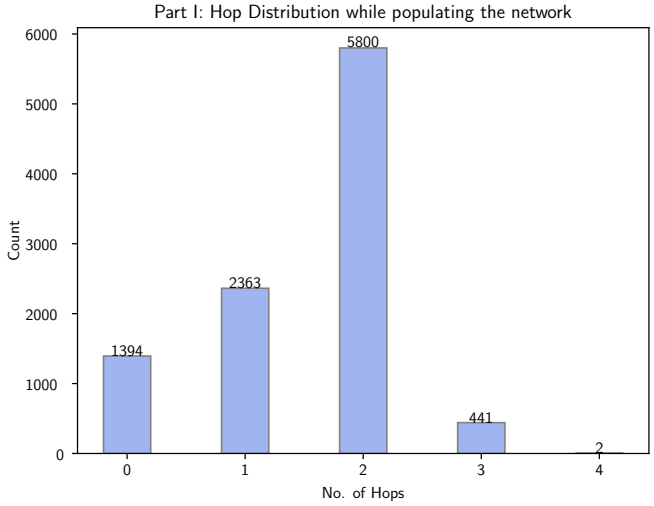

Figure 3: Hop Distribution while populating the network when no of nodes in network is 500.

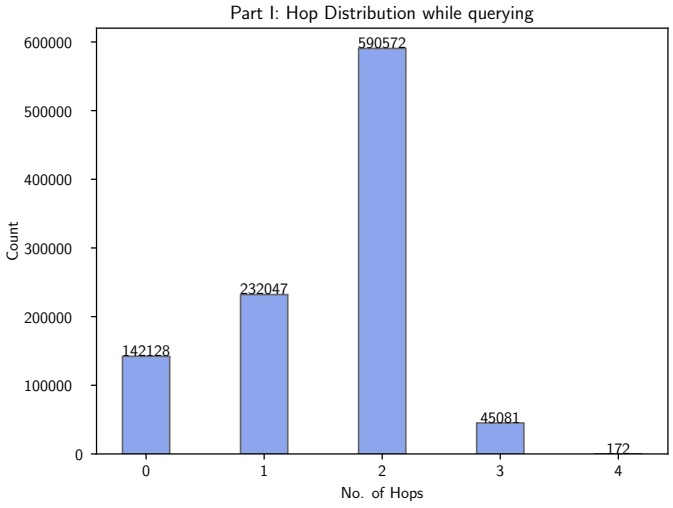

Figure 4: Hop Distribution while querying when no of nodes in network is 500.

No of nodes in network: 1000

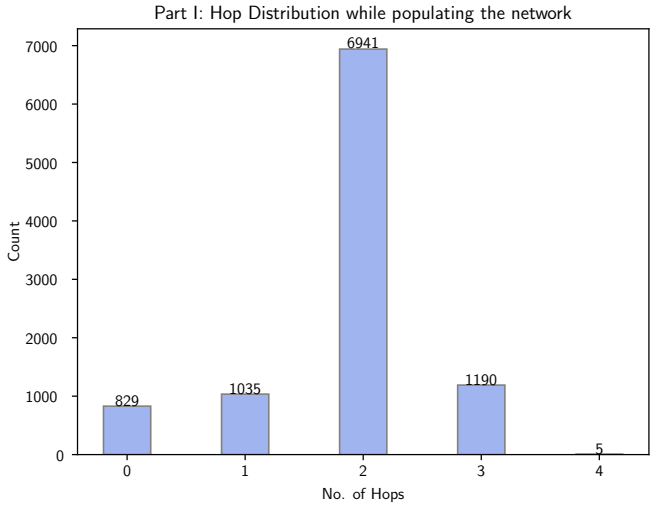

Figure 5: Hop Distribution while populating the network when no of nodes in network is 1000.

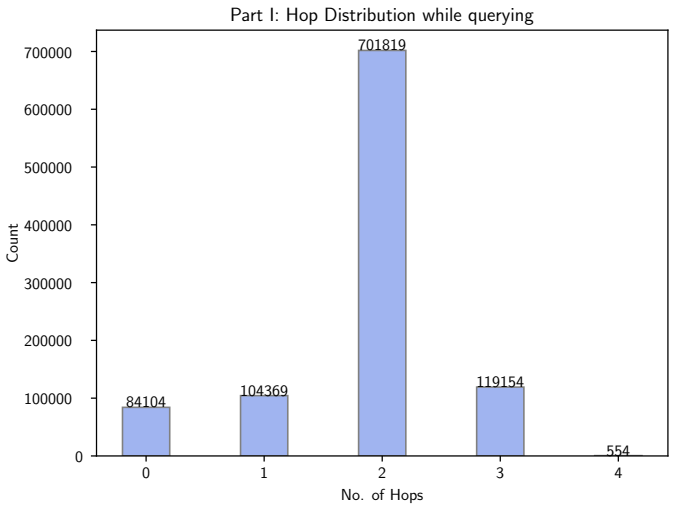

Figure 6: Hop Distribution while querying when no of nodes in network is 1000.

3.3 Evaluation: Part II

In this part we delete half of the nodes presently connected to the network after they have been populated with data. Then we repeat Part I and observe the changes.
After a node is safely deleted its data is transferred to either of present leaf nodes or if none are present they are routed or redistributed to other parts of network.

No of nodes in network: 100

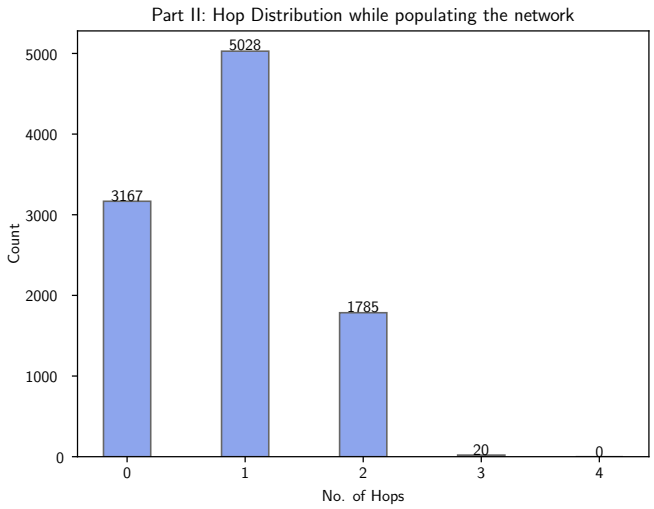

Figure 7: Hop Distribution while populating the network when no of nodes in network is 100.

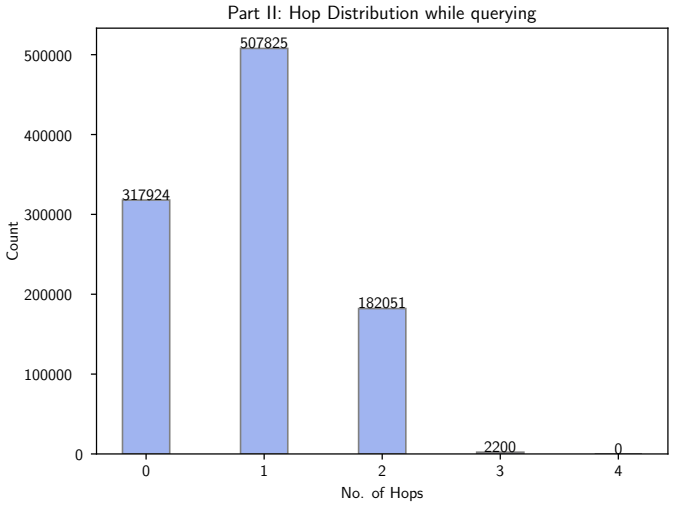

Figure 8: Hop Distribution while querying when no of nodes in network is 100.

No of nodes in network: 500

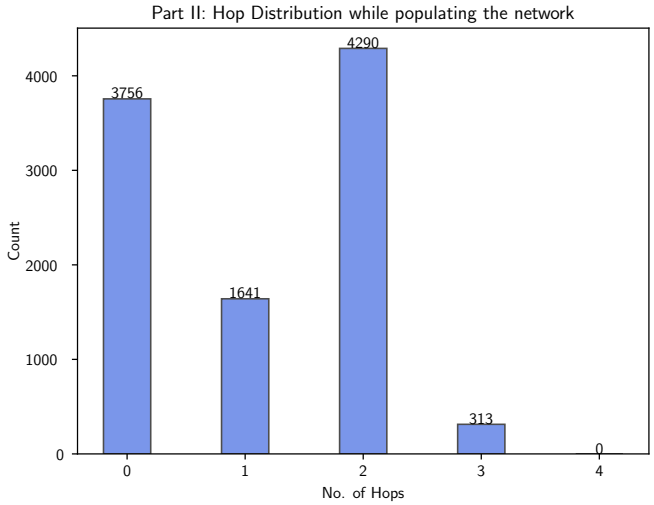

Figure 9: Hop Distribution while populating the network when no of nodes in network is 500.

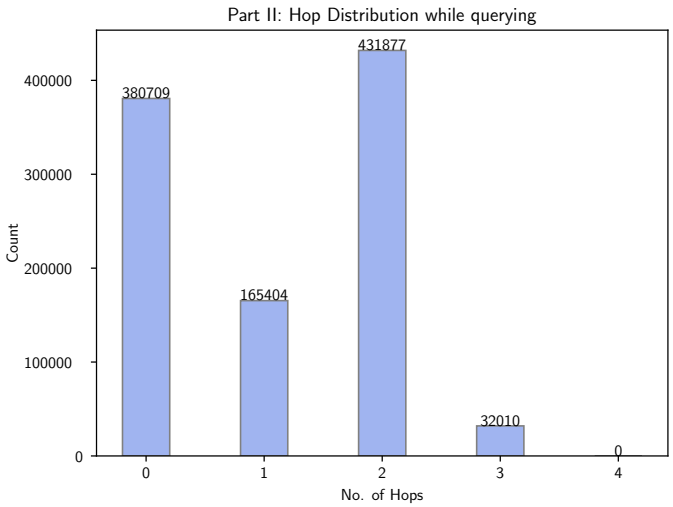

Figure 10: Hop Distribution while querying when no of nodes in network is 500.

No of nodes in network: 1000

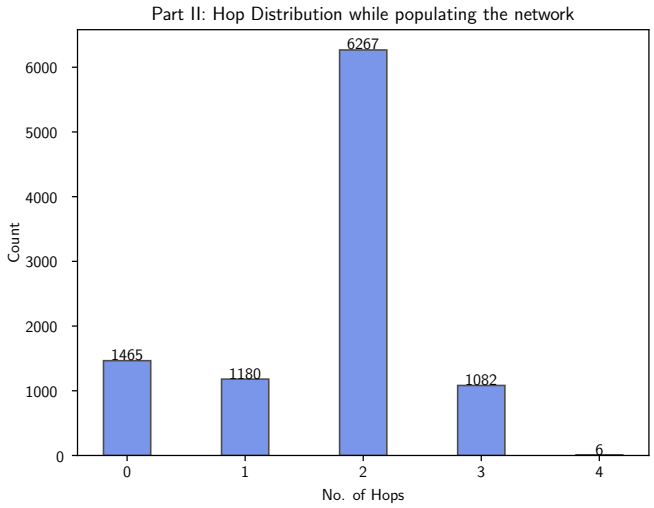

Figure 11: Hop Distribution while populating the network when no of nodes in network is 1000.

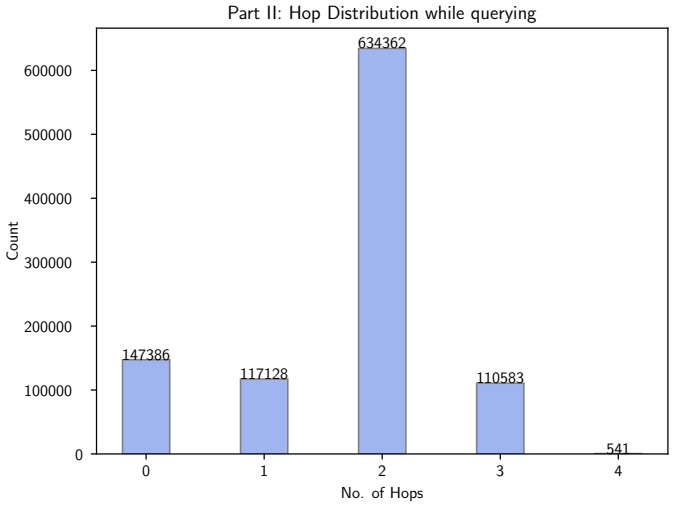

Figure 12: Hop Distribution while querying when no of nodes in network is 1000.

4 OBSERVATION:

4.1 Trends Observed

i. The distribution for number of hops for a particular configuration is almost similar while populating the network and while querying.
ii.There is an increase in no of zero hops while populating and querying in part 2 of exp w.r.t to part 1.

4.2 Justification

For i. Since, the queries we perform are distributed in uniform manner so each data item is queried almost same no of times. So, data which is stored at lets say 3 hops distance will be queried almost same no of times.
For ii. After deletion since the data stored in a node increases, if a node from their surrounding is deleted. So, probability for finding a data on 0 hops increases.

5 INTERFACE:

5.1 How to run:

Shell commands:
$ make
$ ./main

5.2 Interface working:

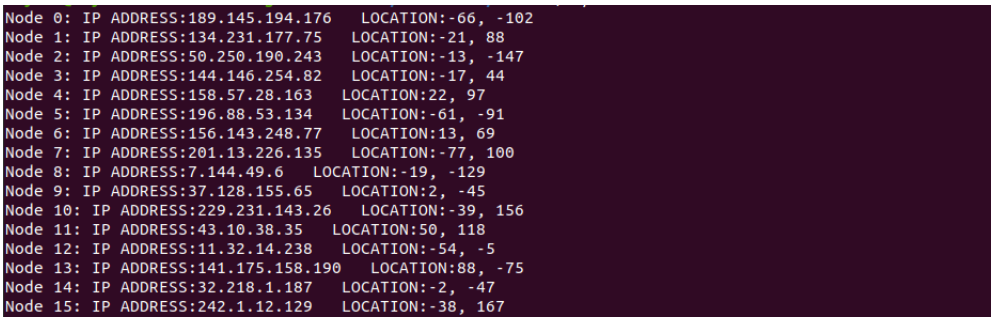

Figure 1: Pastry Initialization.
When the pastry network is initialized it shows the IP address and location of nodes present in network.

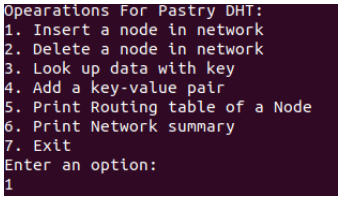

Figure 2: Options Menu.

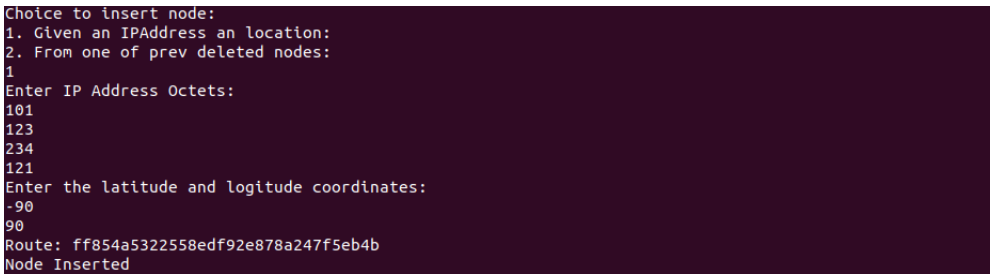

Figure 3: Insertion of a node.

We have to manually input all octets of IP address and longitude and latitude of coordinates of node we want to insert.

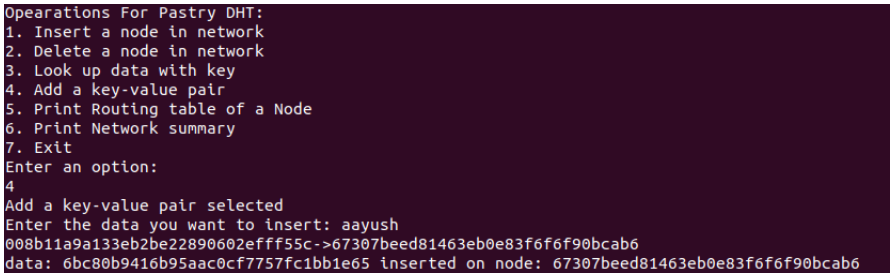

Figure 4: Insertion of a key.

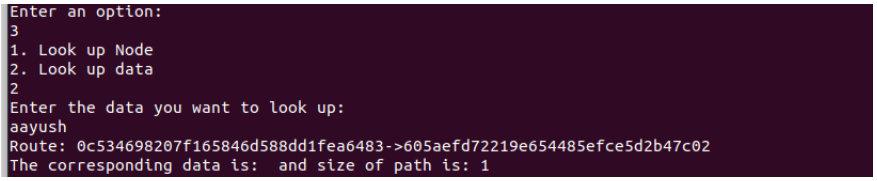

Figure 5: Qurying a key present.

After the key is found it also prints the route of nodes in its path.

5.3 Routing Table with Network Summary:

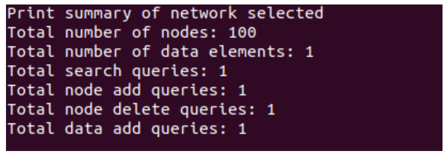

Figure 6: The result of printing the network summary.

It shows the summary of all the operations performed on this network after initialization.

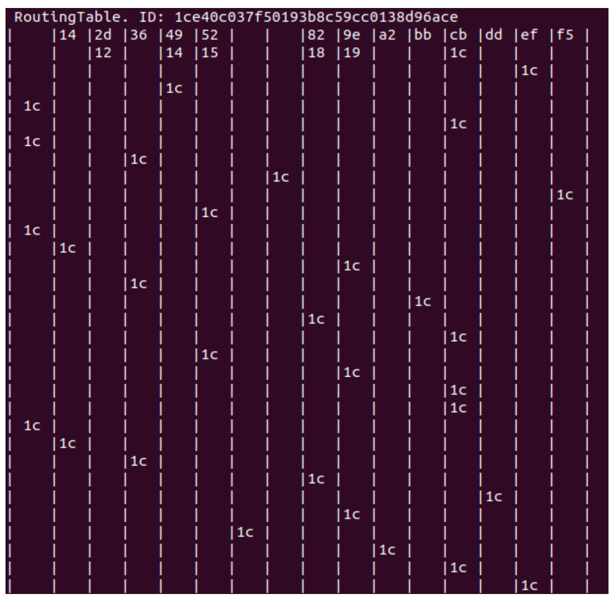

Figure 7: Routing Table of one of the nodes.

Each routing table entry is of 32 characters but due to space contraints is represented using the first 2 characters of it.