Programming Assignment 1: Simple File System on a DHT

Execution Steps

Java Version - 1.7.0_76

Thrift Version - 0.9.3

Compilation Steps:

1) *javac -cp .:<thrift_jars_directory>/* SuperNode.java*
2) *javac -cp .:<thrift_jars_directory>/* Node.java*
3) *javac -cp .:<thrift_jars_directory>/* Client.java*

Steps to make CHORD DHT up and running:

1) *java -cp .:<thrift_jars_directory>/* SuperNode <supernode_port_no>*
2) *java -cp .:<thrift_jars_directory>/* Node <supernode_ip> <supernode_port_no> <node_port_no>*
3) *repeat Step 2 for k-1 number of times, where k = number of nodes you want to add in DHT. Here k <= 32*
4) *java -cp .:<thrift_jars_directory>/* Client <supernode_ip> <supernode_port_no> <operation> <file_name> <contents>*
   operation can be W (write), R (read), I (get DHT Information)
   All the parameters are mandatory for Write.
   <contents> is not required for Read.
   <file_name>, <contents> are not required for DHT Information

Design Architecture

Our System Design has the following main modules/components :

a) SuperNode

SuperNode receives request from clients and nodes of DHT, maintains list of active nodes in DHT and provides list of active nodes. We have added synchronization in SuperNode using trylock, so that no two nodes try to join DHT at same time. We have added supernode service in IDL file of thrift, having following functions - Join, PostJoin, GetNode, getDHTInfo

b) Node

Node adds itself into DHT with help of Supernode. Once it is active in DHT, it receives requests from client for read/write operations. If during read, client has file, it will respond to the client otherwise it will forward request to other nodes based on its finger table (FT). We have explained its functionality in later part of this report

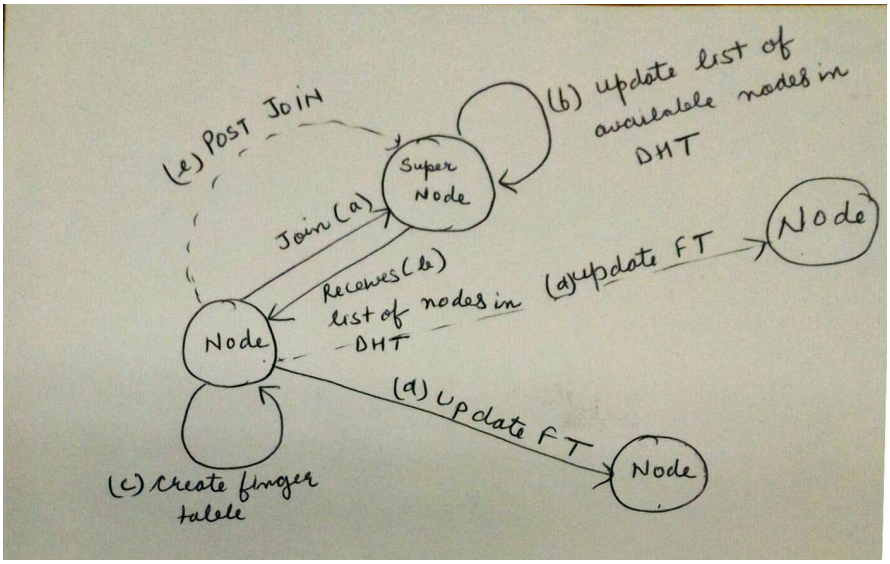
c) Client

Client gets some initial node to contact from Supernode. Once it gets some random node of DHT system, it sends read/write operation to that node which recursively look up file in system and will return appropriate result.

Thrift File

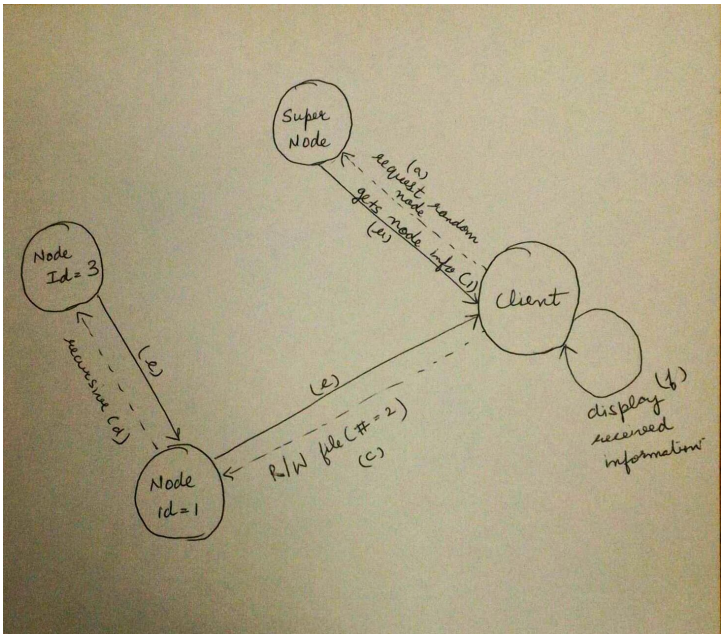
Thrift file contains two service :

a) Node service
b) Supernode Service



Steps of Chord DHT System

In Step a) Node sends a join request to supernode
In Step b) Node receives list of nodes currently in DHT ecosystem from supernode
In Step c) Node builds its own Finger Table (FT)
In Step d) Node Sends Update DHT to all nodes in DHT
In Step e) Node Sends Post join request to SuperNode
In Step f) Super Node updates its list of available nodes in DHT ecosystem



Steps of Read/Write

In Step a) Client requests random node Information from SuperNode
In Step b) Client receives node information(Random node) from SuperNode

In Step c) Client connects to node for which it received information from SuperNode
In Step d) Node checks if it is the destination node by comparing file hash value with its id. If it finds it to be true, it will do read/write operation; otherwise with help of finger table, it forward request to other node.This step is done recursively, until destination node is found.
In Step e) Read/Write Information is returned along the path to client
In Step f) Client displays received information

Assumptions

We have assumed that there are maximum of 32 nodes in the DHT system. Once a node joins a system, it will be there forever. Also, we have assumed that there is zero probability of node failure. We have used simple string hashing to hash filename to integer node. Also, we used random generator for generating node id instead of IP+Port number (We discussed this with TA). Apart from this, one more assumption is first, supernode will come, then Nodes of DHT and then client will make read/write call. It would never happen that after client writes to system, any other node gets added into system. To keep track during file read/write operations, we have also maintained predecessor list

Test Cases

a) One Node in DHT
b) Super Node and Nodes are on same machine and client on remote location
c) Super Node and Client on same machine and Nodes are on remote location
d) SuperNode, Nodes and Client is on Same machine
e) SuperNode, Nodes and Client on different machine
f) Trying to add two nodes at same time

Negative Test Cases:

a) When Firewall is enabled which is blocking port number on client, supernode or nodes machine.
b) When we are using special port numbers such as 8080
c) When we are using port number which are already in use
d) When passing wrong IP address
e) When passing wrong port number
f) Since, we made an assumption of maximum of 32 nodes, what will happen if nodes are more than 32
g) If supernode is down and we are trying to add node in system
h) If supernode is down and client is trying read/write operations
i) If there is no node and client is trying read/write operations

Result:

We successfully implemented CHORD DHT using thrift. Multiple Nodes are able to join DHT system with help of Supernode and successfully updated Finger Table of all nodes in System. Clients are able to read/write different files residing at different locations.

REFERENCES :

i) Thrift RPC tutorial by Kwangsung Oh
ii) Distributed Systems: Principles and Paradigms (2nd Edition)", by Andrew S. Tanenbaum and Maarten van Steen

iii) I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In Proc. ACM SIGCOMM'01, San Diego, CA, Aug. 2001.
iv) Method for generating random numbers (Stack Overflow)
v) Thrift Apache Documentation (https://thrift.apache.org)