README

Huangxin Wang

1. Code

Class	Function
Hashing.java	Hash IP address and user name to ID
CalendarManager.java	Implement the basic function of calendar Implement two phase-commit Implement chord Implement failure detector and failure handler
CalendarClient.java	Call calendarManagerInterface's remote function for user request Implement reconnecting to new server when old server is down
CalendarObj.java	Each users is correspond to a CalendarObj
FingerTable.java	Data structure of finger table
Order.java	Sort clients by id, used for sequential access
Util.java	Utility function, e.g. inBetween(), inBetweenExleft()
Log.java	Used to output debug message

2. ID of node

Here is the id of node and client used in the test. Note for the purpose of testing, we set the keysize of chord very small. In reality, keySize can be set to be enough large to avoid hash conflict.

IP/Name	ID
`129.174.94.81`	27
`129.174.94.82`	23
`129.174.94.91`	9
`129.174.94.87`	18
`129.174.94.86`	11
`129.174.94.98`	14
`Bob`	13
`Sandy`	4
`Jim`	22

3. How to run server

type: . run_server.sh

Note: the server will join to chord by node 9

4. How to run client

type: . 9Client.sh

Note: the client will connect to node 9 to get its successors, note node 9 can change to any other nodes, since every node could find successor for the client. You can also try . 23Client.sh to start client by first connecting to server 23

5. How to test

5.1 Test Chord and Primary copy

1) start server 9, 23, 27
1) start one client end, type: test: test_Bob
   Bob will be added, and an open event will be schedule for Bob
   Bob's primary copy should be in node 23, and its successor should be in node 27
2) start node 18, then primary copy would move from 23 to 18, backup would move from 27 to 23
3) shutdown 23, then backup would move from 23 to 27
4) During all these steps, we can check the finger table of each node.

5.2 Test two phase

1) in client Sandy, type: test: test_twoPhase
   This would schedule an event in which some clients is not exist, therefore, this event will be abort
   Check node 18: it will first VOTE_COMMIT, then doAbort
   Check node 27: it will VOTE_ABORT

5.3 Test Failure of server

Shutdown node 18, then Bob successor would change from 18 to 27

type: myEventList

You will check Bob is reconnect to new server 27. This is transparent to user.