Manual

Running the program:

1. Go to project root (Assignment 2) using Terminal.
2. You can type **"ls"** command to list the files and check whether you are in the correct directory.
3. Type **"make clean"** to delete old compiled .class files.
4. Type **"make"** to compile all project files.
5. Type **"make run"** to run Distributed Hash Table program.
6. Type **"make run_test"** to run Distributed Hash Table test program. (It is used for performance evaluation)
7. Type **"make run_hash_test"** to run the Hash function test pogram. (Used for hash function evaluation)

The document below explains how to use the program, what are the different input the program asks for.

Distributed Hash Table (Peer):

You can run the Peer program by using the makefile command **"make run"**. The Peer program internally loads the configuration file and runs two sub-programs (threads) called Peer Client and Peer Server. Peer Client connects to another Peer Server using port 20000 and the Peer Server works on port 20000.

Let's run the Peer program step-by-step: *(Lines marked in RED are console output.)*

```
make run
```

```
********** PEER CLIENT STARTED **********

********** PEER SERVER STARTED **********

What do you want to do?

1.Add a (key,value) pair.

2.Search for a key.

3.Delete a (key,value) pair.

4.Print log of this peer.

5.Exit.

Enter choice and press ENTER:
```

Enter the respective number choice (1 – 5) of whatever task you want to do. Explanation for each option is as below:

1 = Add (PUT) a <Key, Value> pair in the Distributed Hash Table.

2 = Retrieve (GET) the Value of a Key from Distributed Hash Table.

3 = Remove (DELETE) a <Key, Value> pair from the Distributed Hash Table.

4 = Use this option if you want to check your peer server's log. It is useful if you want to check the various <Key, Value> pairs stored by your peer. Also useful for debugging purpose.

5 = Exit/Terminate the program.

If you select option 1:

Enter KEY (Max. 12 characters):

Enter the Key of the <Key, Value> pair you want to store in the Distributed Hash Table. Key should not be more than 24 bytes i.e. 12 characters in Java.

Enter VALUE (Max. 500 characters):

Enter the Value of the <Key, Value> pair you want to store in the Distributed Hash Table. Value should not be more than 1000 bytes i.e. 500 characters in Java.

The Peer then informs you whether the PUT <Key, Value> pair operation was successful.

If you select option 2:

Enter the KEY which you want to search:

Enter the Key who's Value you want to retrieve from the Distributed Hash Table (DHT).

If the specified Key was found in DHT, the programs displays the Value of the Key. If the Key was not found, then displays the message – "VALUE not found for the given KEY in the Distributed Hash Table."

If you select option 3:

Enter the KEY which you want to delete:

Enter the Key who's <Key, Value> pair you want to remove/delete from the Distributed Hash Table (DHT).

The Peer then informs you whether the DELETE <Key, Value> pair operation was successful.

If you select option 4:

This option prints the peer server log if available. Generally, the Peer server logs various messages while serving requests of other peer clients.

If you select option 5:

This option will close/terminate the application normally. You'll be no longer be in the Distributed Hash Table network. In case a client tries to get/put/delete a <Key, Value> pair from you, it will receive an error message.

Let's run the Distributed Hash Table test program step-by-step: *(Lines marked in RED are console output.)*

You can run the DHT Test program by using the makefile command **“make run_test”**. The Test program can evaluate three operations – PUT, GET and DELETE. It runs the selected operation the specified number of times on specified number of threads in parallel and tells you the time it took to execute all the operations.

Enter the operation you want to test? (PUT, GET, DEL):

Enter either PUT or GET or DELETE depending on the operation you would like to evaluate.

Enter the number of clients you want to test on? (1-8):

Enter the number if threads (integer value) you would like to run in parallel for evaluation. At this time, I am accepting not more than 8 parallel threads. The user should enter between numbers 1 to 8 both including.

Enter the number of operations you want to perform? (Example, 100000):

Enter the number of times (integer value) the specified operation should execute in each thread.

It will take some time to evaluate depending on your evaluation criteria. It displays Total Time in seconds it took to execute the operations, the specified number of times and also the Average Time per operation. If you want to terminate the Test program in between then press **Ctrl + C**.

Let’s run the Hash function test program step-by-step: *(Lines marked in RED are console output.)*

You can run the Hash Function Test program by using the makefile command **“make run_hash_test”**. The Test program will load the text file “rawtext.txt” and get all unique words from it which will be keys for the hash table. It evaluates how dense/sparse is the distribution of keys amongst the peers in the network.

No user input is required to run this test program. Just make sure the file “rawtext.txt” is present in the directory where you are running this program.