**CS 550 – Advanced Operating Systems – Programming Assignment 1**

**Name – Sreeram Haridas**

**CWID – A20330191**

**Napster style P2P system – Performance Evaluation**

**Introduction**  
 The project is aimed at implementing a simple distributed hash table. An example of such a distributed hash table is the ZHT or Zero Hop distributed hash table. Hash tables are used to store and retrieve data in a much efficient when compared to other data retrieval systems. I have used Python as my basic programming language and also used sockets to create connection between the nodes. My program works well with 8 nodes, where each node is a client as well as server. The client enters the key value pair and that is distributed to hash tables across various servers based upon a hash function defined at the client. So the data is distributes across various servers. To retrieve the data, again the hash function is called to find which server has the data and once the call is made data is received. The server configuration is kept constant with the help of configure file and then hence the is no dynamic implementation necessary.

**Performance Evaluation** Benchmarking of code is performed by checking the total time done for the operations put/get/del. The operations are performed in the range of 100K each with 1 client, 2 client, 4 client and 8 client; keeping the 8 servers running already.The total time of each operation is figured out and the time for one job is calculated to benchmark the efficiency of the code.

1. **Test case -1**

Client – 1

Reg/Search/Download – 100K each

|  |  |  |  |
| --- | --- | --- | --- |
| **Test1** | **num of jobs** | **time(s)/Client 1** | **time per job (ms)** |
| REG | 100K | 38.0122 | .3801 |
| SEARCH | 100K | 37.704 | .3770 |
| DOWNLOAD | 100K | 107.9563 | 1.079 |

From the table it can be noted that per Reg/Search/Obt operation took around 1ms and very desirable for a well-functioning system with a CIS.

1. **Test case -2**

Client – 2

REG/SEARCH/DOWNLOAD – 100K each

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test2** | **num of jobs** | **time(s)/Client 1** | **time per job (ms)** | **time(s)/Client 2** | **time per job (ms)** |
| REG | 100K | 50.23 | 0.5023 | 54.092 | 0.5409 |
| SEARCH | 100K | 61.84 | 0.6184 | 64.13 | 0.6413 |
| Download | 100K | 221.7113 | 2.217 | 241.0313 | 2.4103 |

1. **Test case -3**

Client – 4

REG/SEARCH/DOWNLOAD – 100K each

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test3** | **num of jobs** | **time(s)/Client 1** | **time per job (ms)** | **time(s)/Client 2** | **time per job (ms)** | **time(s)/Client 3** | **time per job (ms)** | **time(s)/Client 4** | **time per job (ms)** |
| REG | 100K | 68.7459 | 0.6874 | 72.5639 | 0.7256 | 66.8536 | 0.6685 | 71.124 | 0.7112 |
| SEARCH | 100K | 79.78 | 0.7978 | 77.819 | 0.7781 | 81.1024 | 0.81102 | 80.4071 | .8040 |
| DOWNLOAD | 100K | 548.1839 | 5.4810 | 578.8558 | 5.7285 | 584.5336 | 5.845 | 571.7179 | 05.7171 |

1. **Test case -4**

Client – 8

REG/SEARCH/DOWNLOAD – 100K each

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test4** | **num of jobs** | **time(s)/Client 1** | **time per job (ms)** | **time(s)/Client 2** | **time per job (ms)** | **time(s)/Client 3** | **time per job (ms)** | **time(s)/Client 4** | **time per job (ms)** | **time(s)/Client 5** | **time per job (ms)** | **time(s)/Client 6** | **time per job (ms)** | **time(s)/Client 7** | **time per job (ms)** | **time(s)/Client 8** | **time per job (ms)** |
| REG | 100K | 114.038 | 1.14 | 105.559 | 1.05 | 111.8005 | 1.11 | 107.2441 | 1.07 | 106.2501 | 1.03 | 111.9102 | 1.11 | 109.63 | 1.09 | 111.6711 | 1.11 |
| SEARCH | 100K | 148.83 | 1.48 | 153.959 | 01.53 | 151.355 | 1.51 | 157.1882 | 1.57 | 152.5159 | 1.52 | 150.0159 | 1.5001 | 153.76 | 1.53 | 151.1937 | 1.519 |
| DOWNLOAD | 100K | 1178.184 | 11.78 | 1153.959 | 11.53 | 1251.07 | 12.51 | 1201.959 | 12.01 | 1284.969 | 12.84 | 1266.026 | 12.66 | 1198.256 | 11.98 | 1196.702 | 11.967 |