Performance Document

Name: Ning Zhang
CWID: A20336916
Department of Computer Science, Illinois Institute of Technology

Firstly, we introduce the method of testing the performance. We design a test function in the source code, which respectively runs 100K PUT commands, 100K GET commands and 100K DELETE commands, then respectively calculating the response time per one kind of request. And at first we deploy 8 clients and servers as Fig1.

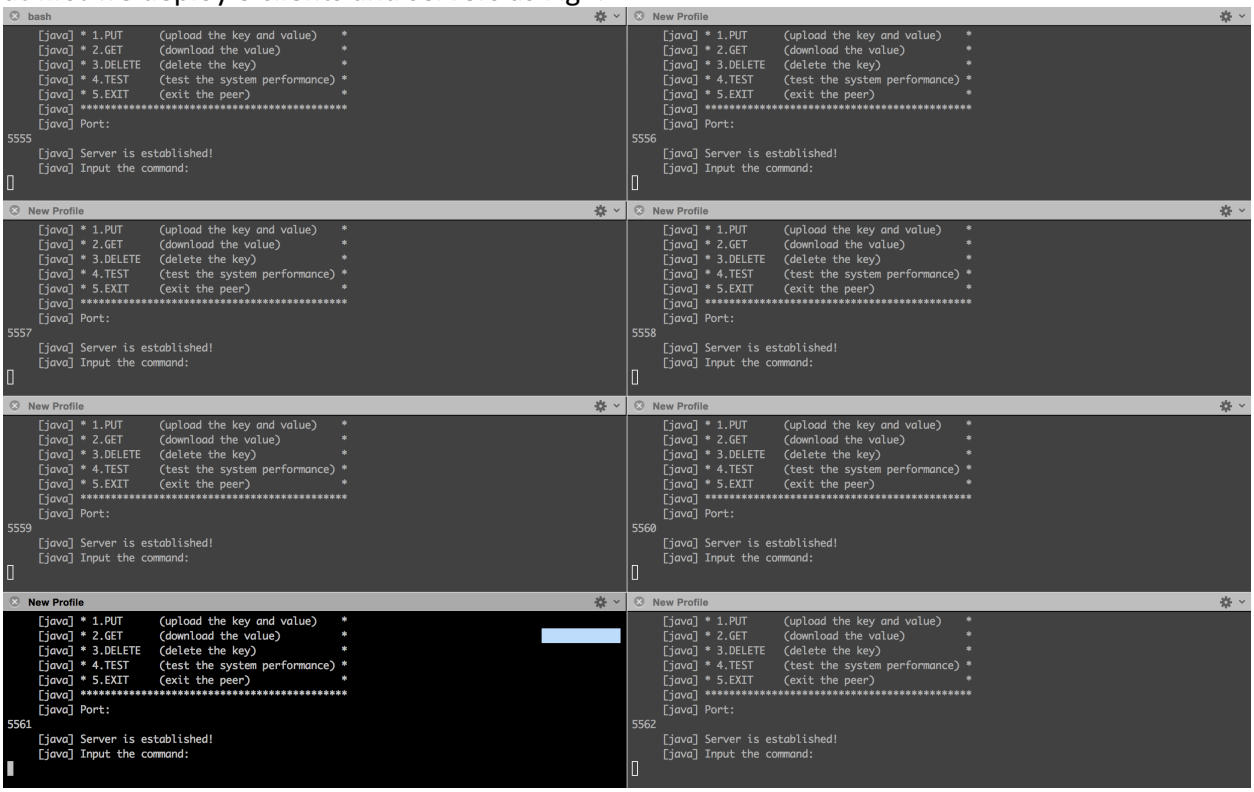

Fig1. Deploying 8 clients and servers

Then we make 1 client run the test function and output the response time of the three kinds of requests as Fig2.

```
file1
     [java] the value is CS550PA2
     [java] Input the command:
DELETE
     [java] key:
file1
     [java] Input the command:
GET
     [java] key:
file1
     [java] the key doesn't exist!
     [java] Input the command:
TEST
     [java] Start performance test, please be patient ...
     [java] put test takes: 63.96 us
     [java] get test takes: 55.739999999999995 us
     [java] delete test takes: 53.05 us
     [java] Input the command:
```

Fig2. one client runs the test command concurrently

And we make two clients run the test function concurrently by using the terminal broadcasting input to make the two terminals concurrently run the test command. Then we get the output results as Fig3.

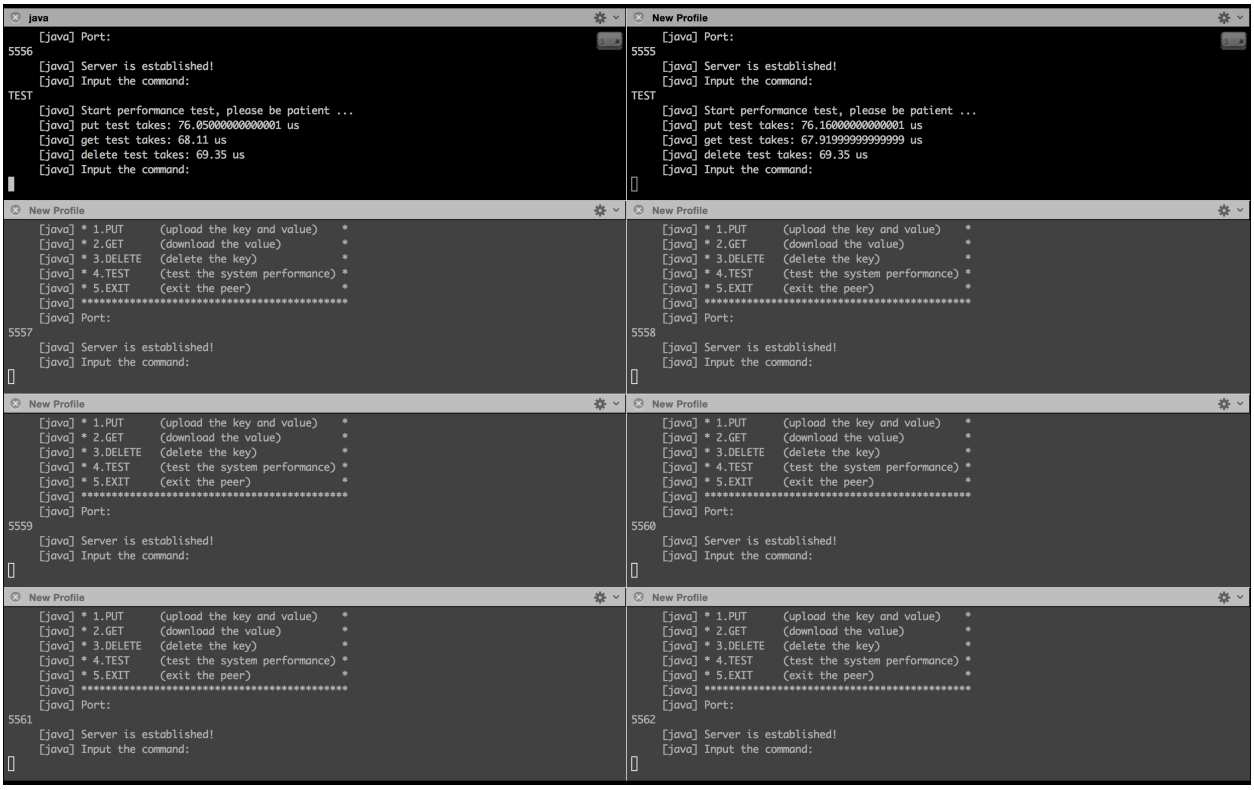

Fig3. two clients run the test command concurrently

And we make four clients run the test function concurrently by using the terminal broadcasting input to make the four terminals concurrently run the test command. Then we get the output results as Fig4.

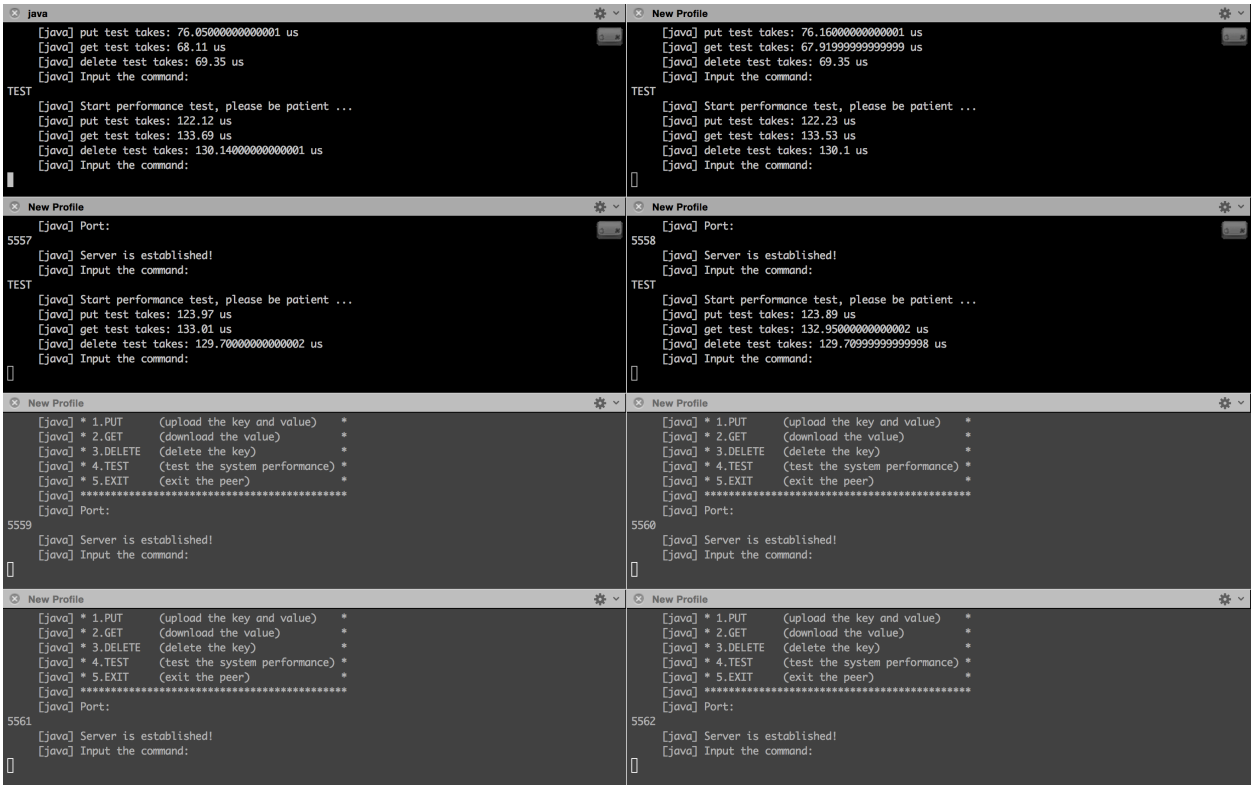

Fig4. four clients run the test command concurrently

And we make eight clients run the test function concurrently by using the terminal broadcasting input to make the eight terminals concurrently run the test command. Then we get the output results as Fig5.

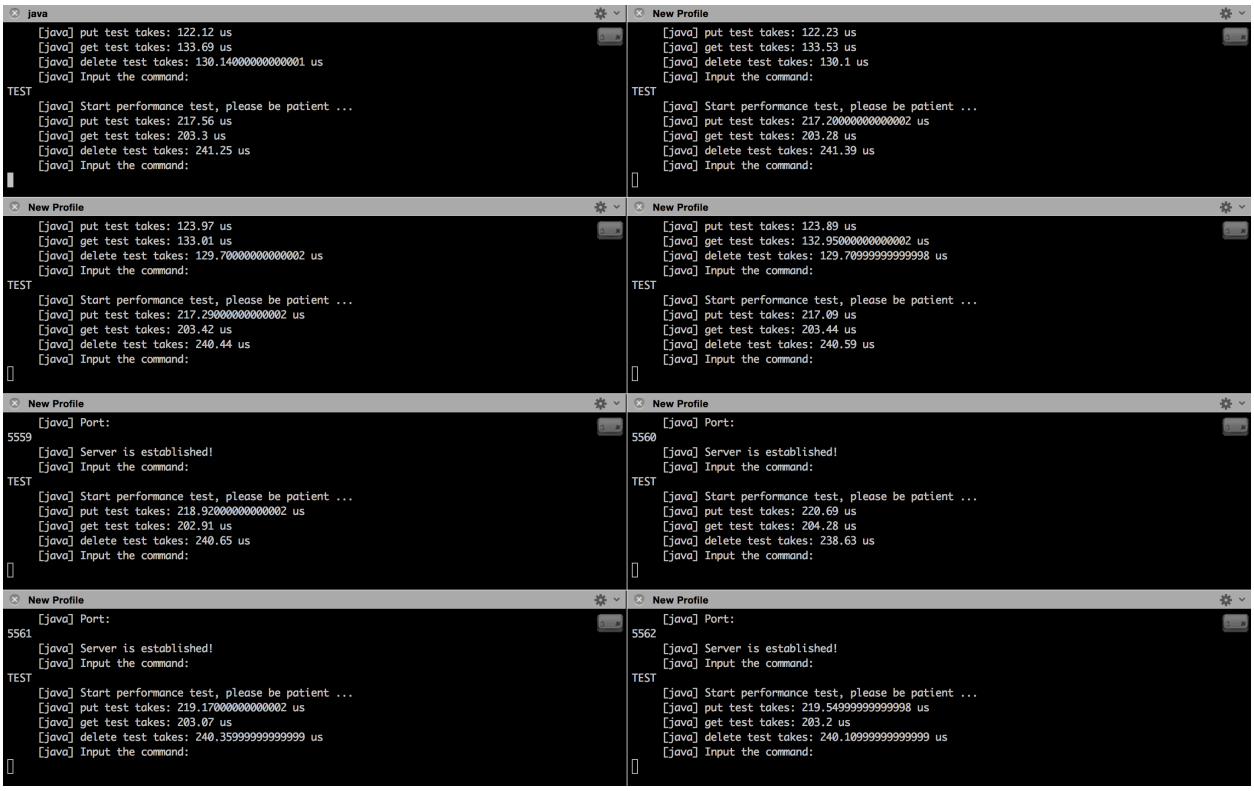

Fig5. eight clients run the test command concurrently

After we get the results of all outputs above, we calculate the average response time per client, then we can get the Fig 6. Performance of Distributed Hash Table. X-axis is the response time per request. Y-axis is the number of clients running concurrently. The blue bar is the response time of one PUT operation. The orange bar is the response time of one GET operation. The gray bar is the response time of one DELETE operation. We can see that the more clients run concurrently, the more response time is spent by the operation. It is because the more clients run concurrently, the more source of the pc is spent. And the more clients run concurrently, the network has more congestion. And the competition will increase when the more clients run concurrently. So the response time of one operation increases as the number of clients running concurrently increases. When there is one client running, the PUT operation cost the most time, the GET and DELETE operation almost cost the same. Because the PUT operation includes two write operations to the hash table, the GET operation includes one read operation to the hash table and the DELETE operation includes one write operation.

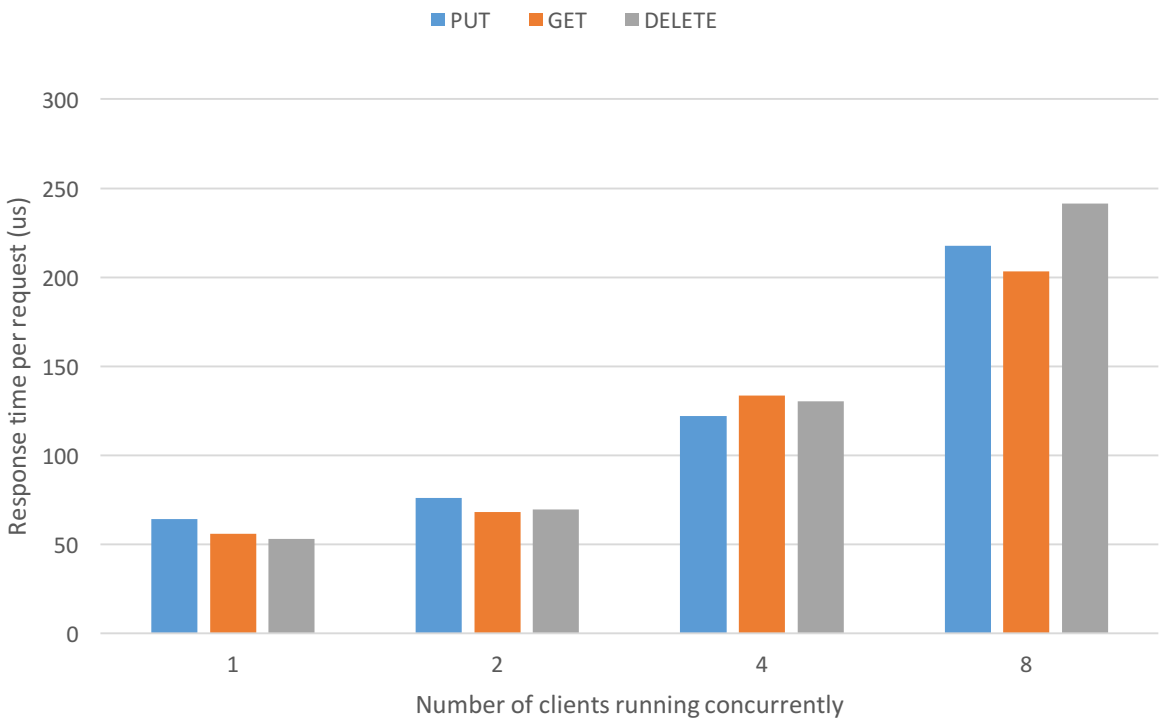

Fig6. Performance of Distributed Hash Table