

Performance Evaluation

Test environment:

- Processor : Intel Core i5
- RAM : 6GB
- Server and clients running on the same system using different ports

Test cases

1. **Test 1:** Running Single client with 100000 requests. The unit of time is ms

	Put	Get	Delete
Client1	2943	2410	2098

Average response time per put request = $2943 / 100000 = 0.02943$ ms

Average response time per get request = $2410 / 100000 = 0.02410$ ms

Average response time per delete request = $2098 / 100000 = 0.02098$ ms

2. **Test 2:** Running two clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	2631	2597	2260
Client2	2585	2623	2229
Avg	2608	2610	2244.5

Average response time per put request = $2608 / 100000 = 0.02608$ ms

Average response time per get request = $2610 / 100000 = 0.02610$ ms

Average response time per delete request = $2244.5 / 100000 = 0.02244$ ms

3. **Test 3:** Running three clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	3786	3868	2869
Client2	3838	3806	3153
Client3	4104	3658	3353
Avg	3909.3	3777.3	3125

Average response time per put request = $3909.3 / 100000 = 0.03909$ ms

Average response time per get request = $3777.3 / 100000 = 0.03777$ ms

Average response time per delete request = $3125 / 100000 = 0.03125$ ms

4. Test 4: Running four clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	4022	4570	3927
Client2	4398	4621	3978
Client3	4718	4596	3305
Client4	4332	4537	3698
Avg	4367.5	4581	3727

Average response time per put request = $4022 / 100000 = 0.04022$ ms

Average response time per get request = $4570 / 100000 = 0.04570$ ms

Average response time per delete request = $3927 / 100000 = 0.03927$ ms

5. Test 5: Running five clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	5543	4518	4582
Client2	5400	5211	5014
Client3	6259	5653	4260
Client4	5787	5363	4303
Client5	5313	5865	3947
Avg	5660.4	5322	4421.2

Average response time per put request = $5660.4 / 100000 = 0.05660$ ms

Average response time per get request = $5322 / 100000 = 0.05322$ ms

Average response time per delete request = $4421.2 / 100000 = 0.04421$ ms

6. Test 6: Running six clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	6000	5887	5617
Client2	6840	6080	5362
Client3	7848	5860	4813
Client4	6210	6330	4906
Client5	6812	6154	4204
Client6	6661	6316	3964
Avg	6728.5	6104.5	4811

Average response time per put request = $6728.5 / 100000 = 0.06728$ ms

Average response time per get request = $6104.5 / 100000 = 0.06104$ ms

Average response time per delete request = $4811 / 100000 = 0.04811$ ms

7. Test 7: Running seven clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	6314	6429	7103
Client2	8060	6878	6814
Client3	9032	7629	5190
Client4	7943	6979	5735
Client5	6744	6874	6554
Client6	8146	7963	4508
Client7	8037	7926	4078
Avg	7753.7	7239.7	5711.7

Average response time per put request = $7753.7 / 100000 = 0.07753$ ms

Average response time per get request = $7239.7 / 100000 = 0.07239$ ms

Average response time per delete request = $5711.7 / 100000 = 0.05711$ ms

8. Test 8: Running eight clients simultaneously with 100000 requests.

	Put	Get	Delete
Client1	8151	8652	6923
Client2	8480	8641	7626
Client3	8767	8544	7046
Client4	9712	7662	6790
Client5	9781	8607	4914
Client6	9885	8763	4208
Client7	7067	8944	5547
Client8	8216	8465	4506
Avg	8757.4	8534.8	5945

Average response time per put request = $8757.4 / 100000 = 0.08757 \text{ ms}$

Average response time per get request = $8534.8 / 100000 = 0.08534 \text{ ms}$

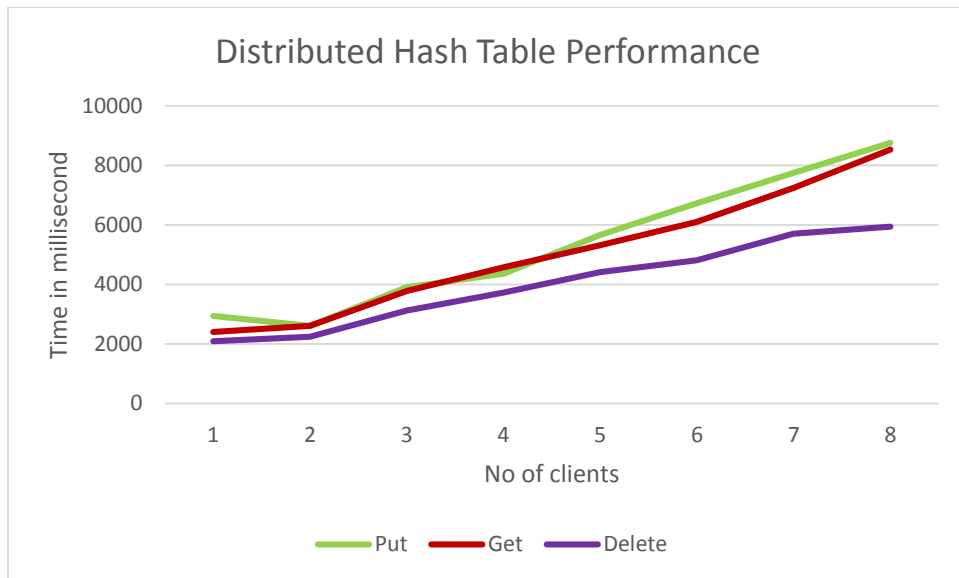
Average response time per delete request = $5945 / 100000 = 0.05945 \text{ ms}$

Performance of the System

Below is the performance of the system with increasing number of simultaneous clients

No of Clients	Put	Get	Delete
1	2943	2410	2098
2	2608	2610	2244.5
3	3909.3	3777.3	3125
4	4367	4581	3727
5	5660.4	5322	4421.2
6	6728.5	6104.5	4811
7	7753.7	7239.7	5711.7
8	8757.4	8534.7	5945

Below is graphical representation of the response time



Conclusion:

The response time is increasing with the increasing number of clients simultaneously requesting the servers.

Test Case Implementation

100K put requests:

```
private void testPut(int numOfReqs) {
    System.out.println("Put test started");
    System.out.println("Wait for the result...");
    long startTime = System.currentTimeMillis();
    for (int i=0; i< numOfReqs; i++){
        String key = "k" + i;
        String value = "Value" + i;
        distributedHashTable.put(key, value);
    }
    long endTime = System.currentTimeMillis();
    System.out.println("Time taken for "+numOfReqs+" puts : " + (endTime - startTime) + "ms");
}
```

100K get requests:

```
private void testGet(int numOfReqs) {
    System.out.println("Get test started");
    System.out.println("Wait for the result...");
    long startTime = System.currentTimeMillis();
    for (int i=0; i< numOfReqs; i++){
        String key = "k" + i;
        distributedHashTable.get(key);
    }
    long endTime = System.currentTimeMillis();
    System.out.println("Time taken for "+numOfReqs+" get requests : " + (endTime - startTime) + "ms");
}
```

100K delete requests:

```
private void testDelete(int numOfReqs) {
    System.out.println("Get test started");
    System.out.println("Wait for the result...");
    long startTime = System.currentTimeMillis();
    for (int i=0; i< numOfReqs; i++){
        String key = "k" + i;
        distributedHashTable.delete(key);
    }
    long endTime = System.currentTimeMillis();
    System.out.println("Time taken for "+numOfReqs+" delete requests : " + (endTime - startTime) + "ms");
}
```