Performance Evaluation

Linlin Chen

A20348195

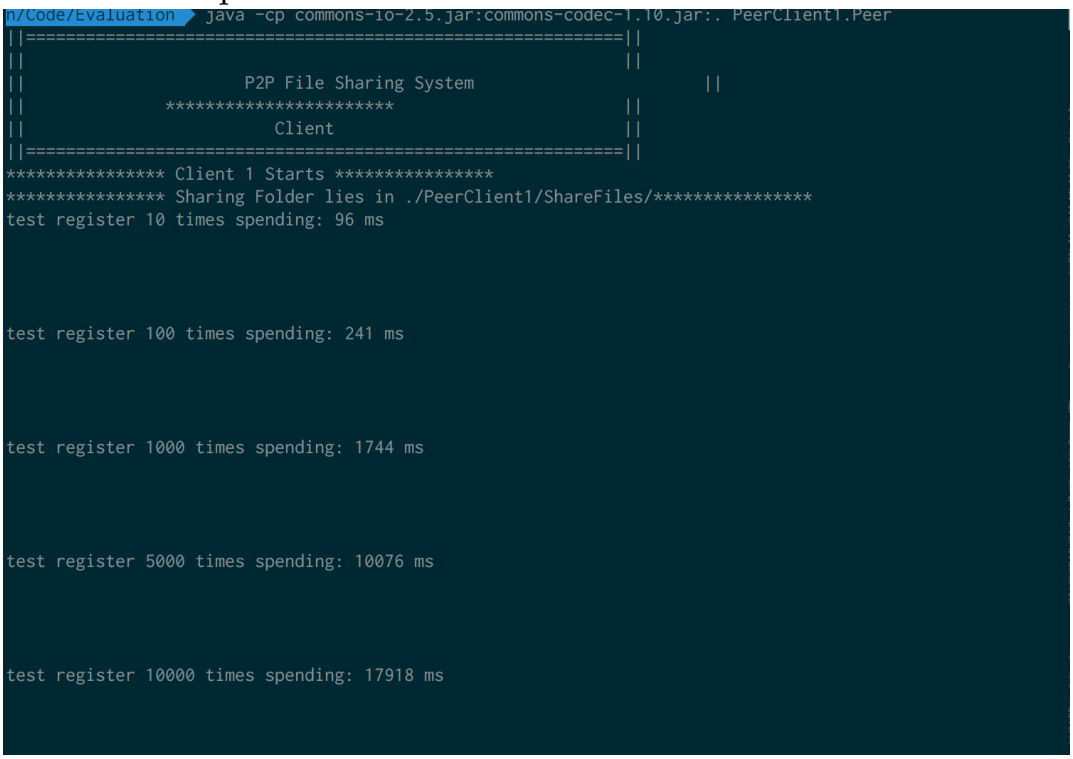
Lchen96@hawk.iit.edu

In this report, I will mainly evaluate the response time. Other evaluations have been done in OutputFile and ManualFile.
To run the performance code, please refer the Evaluation folder. Some codes have been modified for better evaluation convenience.

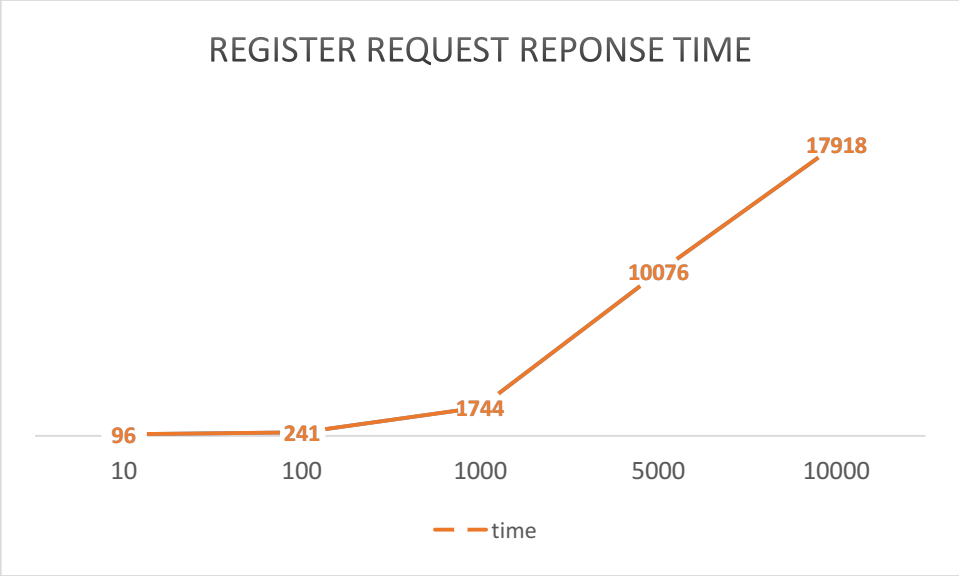
One client scenario:

1. Register response

I test how long would a client get response when sending register request to server. I set the register request times as 10, 100, 1000, 5000 and 10000, and record the overall time below. I will calculate the average time for one request.

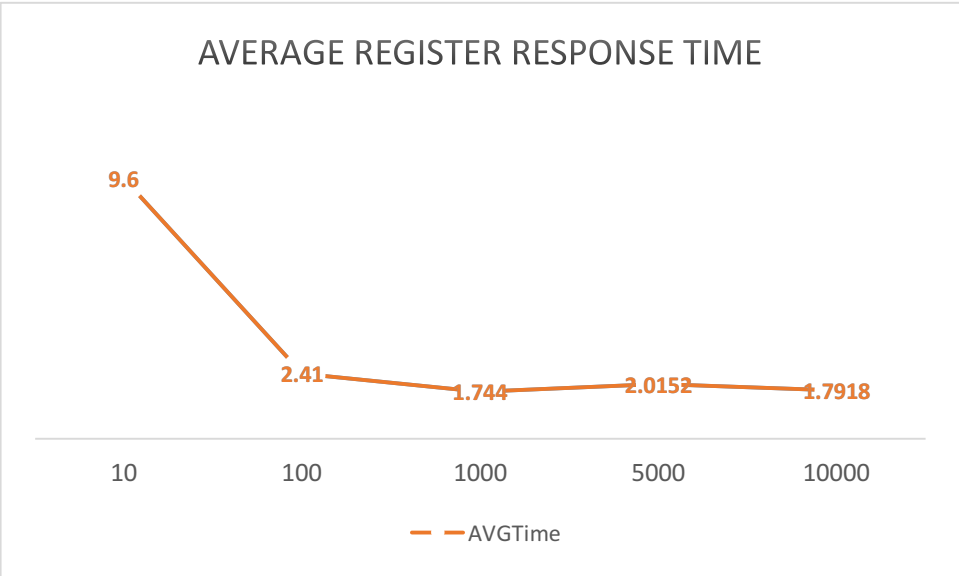


numers	10	100	1000	5000	10000
time/ms	96	241	1744	10076	17918



Then I calculate the average time response time. We can see more requests actually degrade the request time, which should be a good thing.

Numers	10	100	1000	5000	1000(
AVGTime/ms	9.6	2.41	1.744	2.0152	1.791

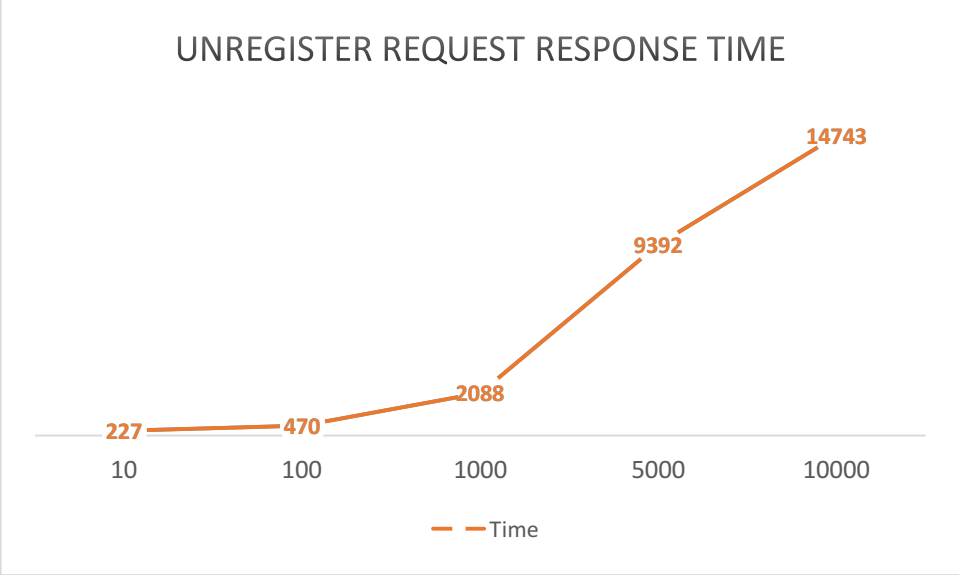


2. Unregister response

I test how long would a client get response when sending unregister request to server. The setting is the same with register test, again I set the register request times as 10, 100, 1000, 5000 and 10000, and record the overall time below.

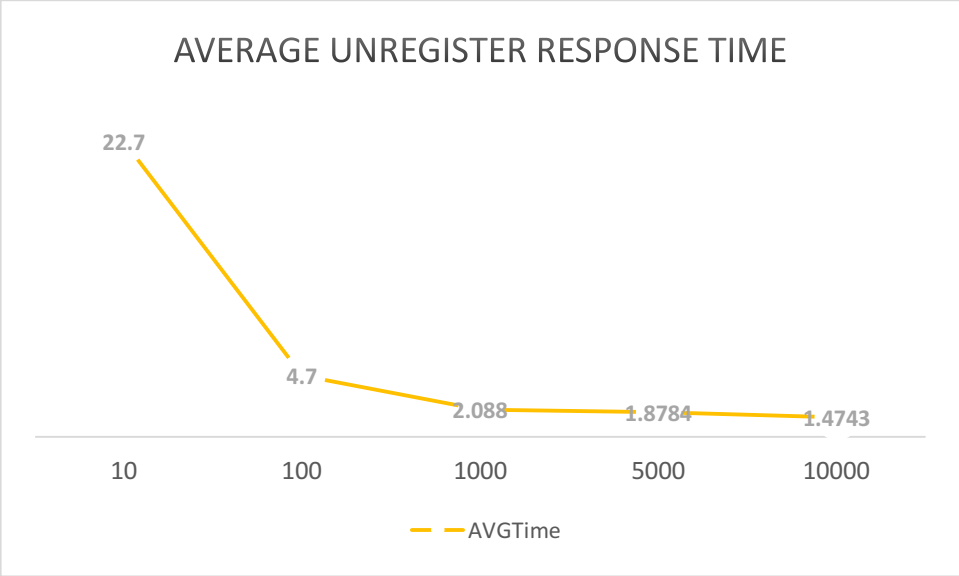
```
n/Code/Evaluation  java -cp commons-io-2.5.jar:commons-codec-1.10.jar:. PeerClient1.Peer
||=============================================================||
||                                                             ||
||                      P2P File Sharing System                ||
||              **********************                         ||
||                         Client                              ||
||=============================================================||
**************** Client 1 Starts ****************
**************** Sharing Folder lies in ./PeerClient1/ShareFiles/****************
test unregister 10 times spending: 227 ms



test unregister 100 times spending: 470 ms



test unregister 1000 times spending: 2088 ms



test unregister 5000 times spending: 9392 ms



test unregister 10000 times spending: 14743 ms
```

Numers	10	100	1000	5000	10000
AVGTime/ms	227	470	2088	9392	14743



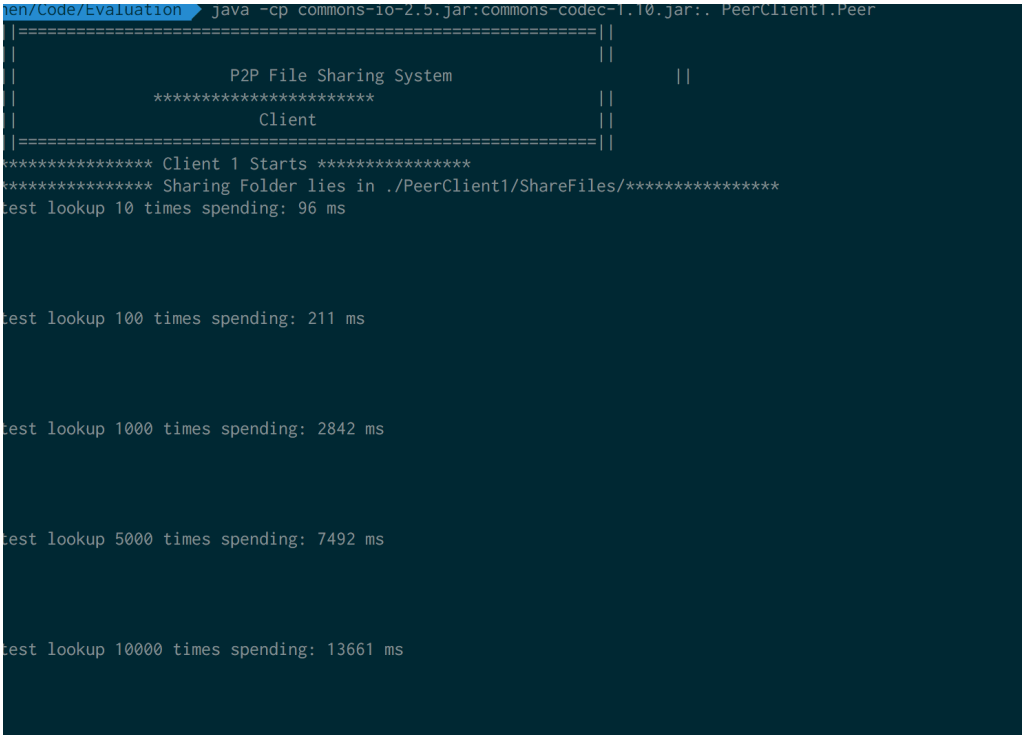
I also calculate the average time below:

Numers	10	100	1000	5000	10000
AVGTime/ms	22.7	4.7	2.088	1.8784	1.4743

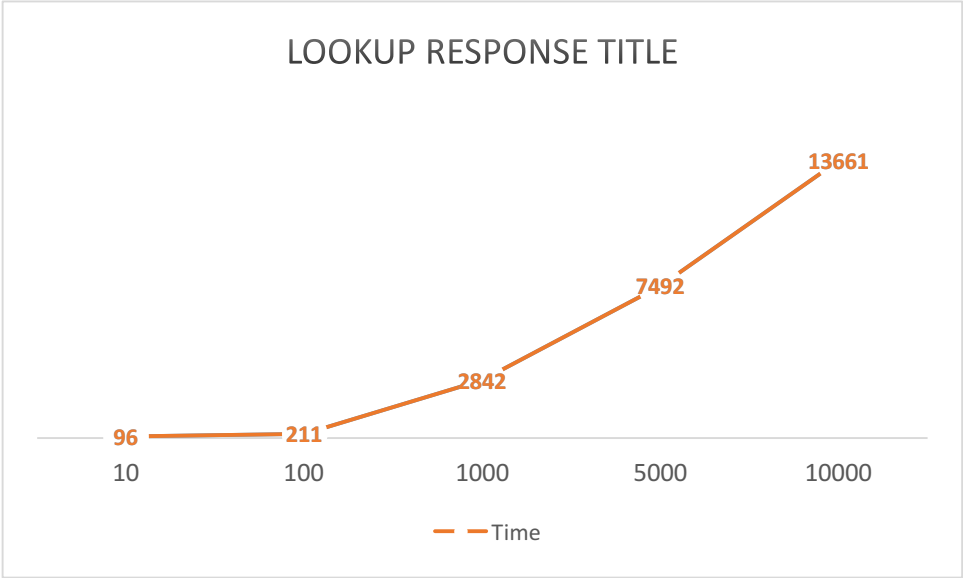


3. Lookup response

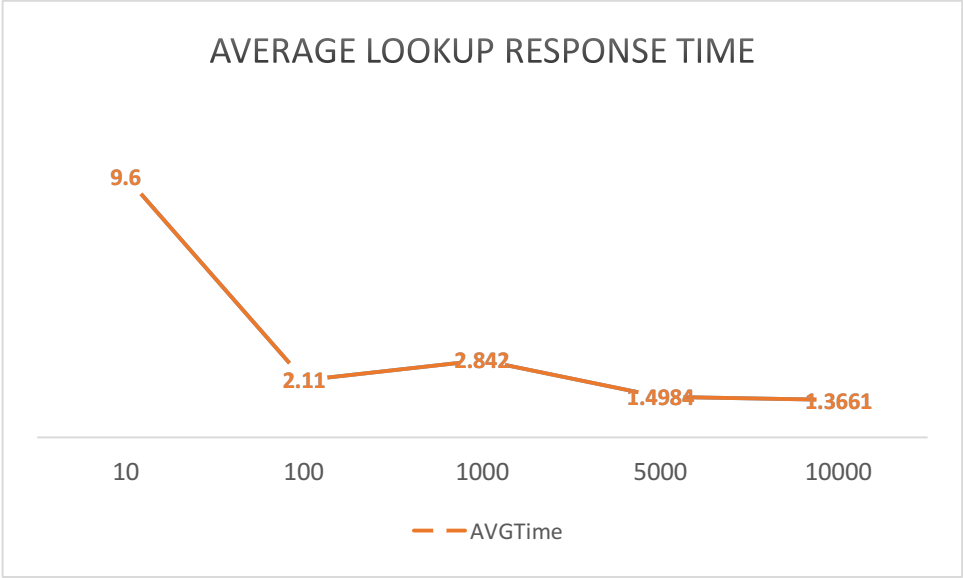
The setting here is the same. I set the lookup request times as 10, 100, 1000, 5000 and 10000, and record the overall time below.



Numers	10	100	1000	5000	10000
Time/ms/ms	96	211	2842	7492	13661

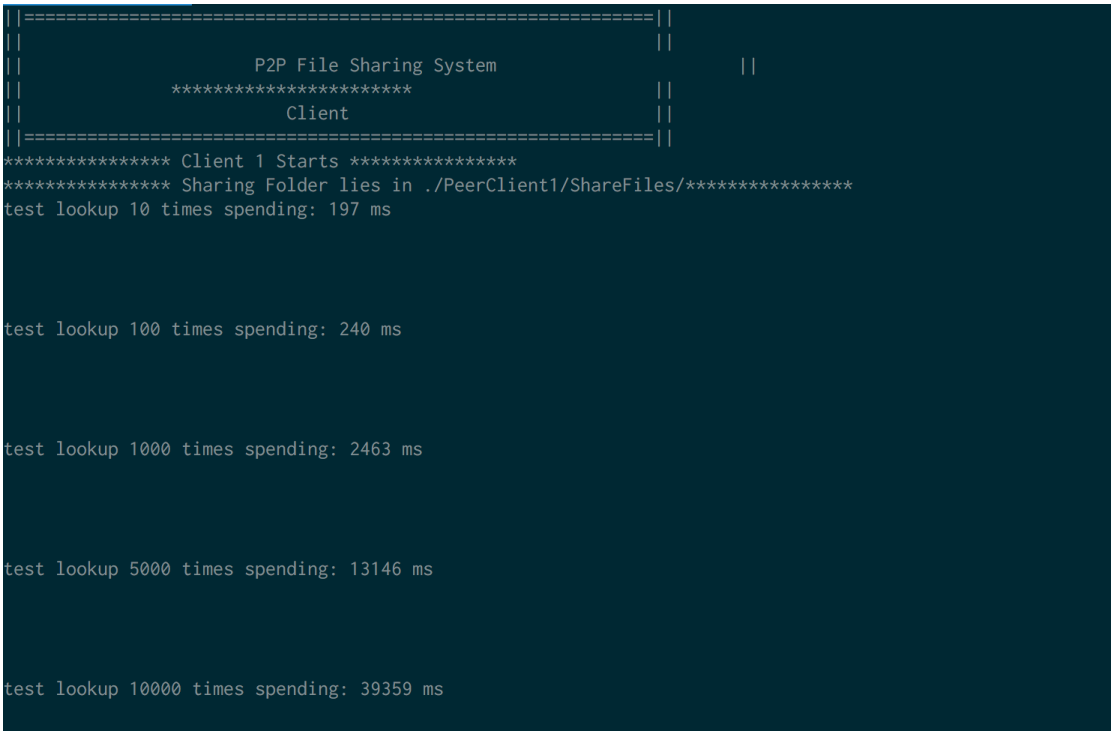


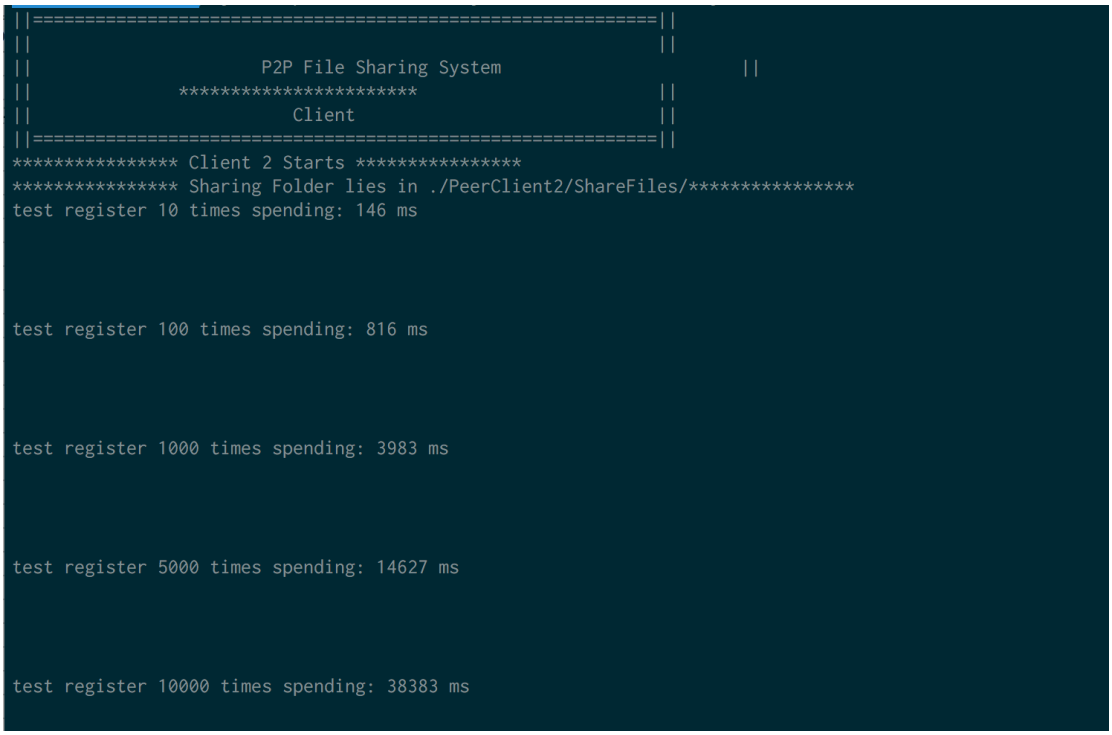
Numers	10	100	1000	5000	10000
AVGTime/ms	22.7	4.7	2.088	1.8784	1.4743



Multiple Client Scenario

I also test when multiple clients concurrently send request, unregister and lookup, how would be the response time changed. I show one running result below. For the simplify, I will not show other running results screenshots.





I calculate modify the client number from 1 to three, and show the average time per client per request. (The summation of all time divided by the client number and request number).

1. Register request time

	Register Request numbers					
Client Num	AVGTime/ms	10	100	1000	5000	10000
	1	9.6	2.41	1.744	2.0152	1.7918
	2	19.3	5.31	4.211	4.98	3.122
	3	24.1	9.321	8.46	6.343	5.43

We can see with more clients, the average time increases, approximately the same order with the client number.

2. Unregister request time

	Unregister Request numbers					
Client Num	AVGTime/ms	10	100	1000	5000	10000
	1	8.6	3.21	2.131	2.253	1. 938
	2	15.91	5.246	4.421	5.233	3.41
	3	25.562	9.981	8.12	7.445	4.912

Again, we can see the with more clients the average time increases, and with more requests, the average time decrease. The increasing of time shares the same order with the increasing of clients' number.

	Unregister Request numbers					
Client Num	AVGTime/ms	10	100	1000	5000	1000
	1	19.1	5.214	3.546	3.861	2.90
	2	29.912	6.611	4.977	6.1223	5.60
	3	35.785	9.88	9.014	8.991	6.12

3. Lookup request time

We can also find that the with more clients the average time increases, and with more requests, the average time decrease. The increasing of time shares the same order with the increasing of clients' number.