CS 342 – Operating Systems

Project 2

Seyfullah Yamanoğlu

1. Environment

We have conducted experiments of our report on Ubuntu Desktop 64-bit 16.04.3 LTS running on virtual machine with 4 GB RAM, 20GB dynamically allocated memory and 1 CPU.

2. Experimets

The test are conducted with 3 different client numbers {1, 5, 10, 15, 20} each with 3 different input sizes {300, 3.000, 30.000} lines. Each line contains 1024 bytes of characters, 256 characters including newline character in the end. Experiment results are taken in milliseconds.

a. 1 Client

	Number of lines for the input file		
Runs	500	5.000	50.000
1.	0.3	71.5	863.7
2.	0.2	84.1	860.7
3.	0.2	71.9	904.4
4.	0.2	72.1	819.0
5.	0.2	75.9	899.9
6.	0.2	68.1	867.2
7.	0.2	79.9	940.5
8.	0.2	100.9	891.7
9.	0.2	89.3	851.3
10.	0.2	81.1	872.4
Average	0.21	79.48	877.08

b. 5 Clients

	Number of lines for the input file		
Runs	500	5.000	50.000
1.	42.9	826.4	8163.1
2.	48.0	751.6	8150.6
3.	40.0	874.8	7718.3
4.	52.3	777.7	8247.7
5.	39.5	717.6	7875.7
6.	58.8	763.8	8180.1
7.	48.5	780.1	8255.9
8.	48.0	652.2	8565.1
9.	4.2	781.0	7587.8
10.	35.8	700.2	8248.2
Average	41.8	762.54	8099.25

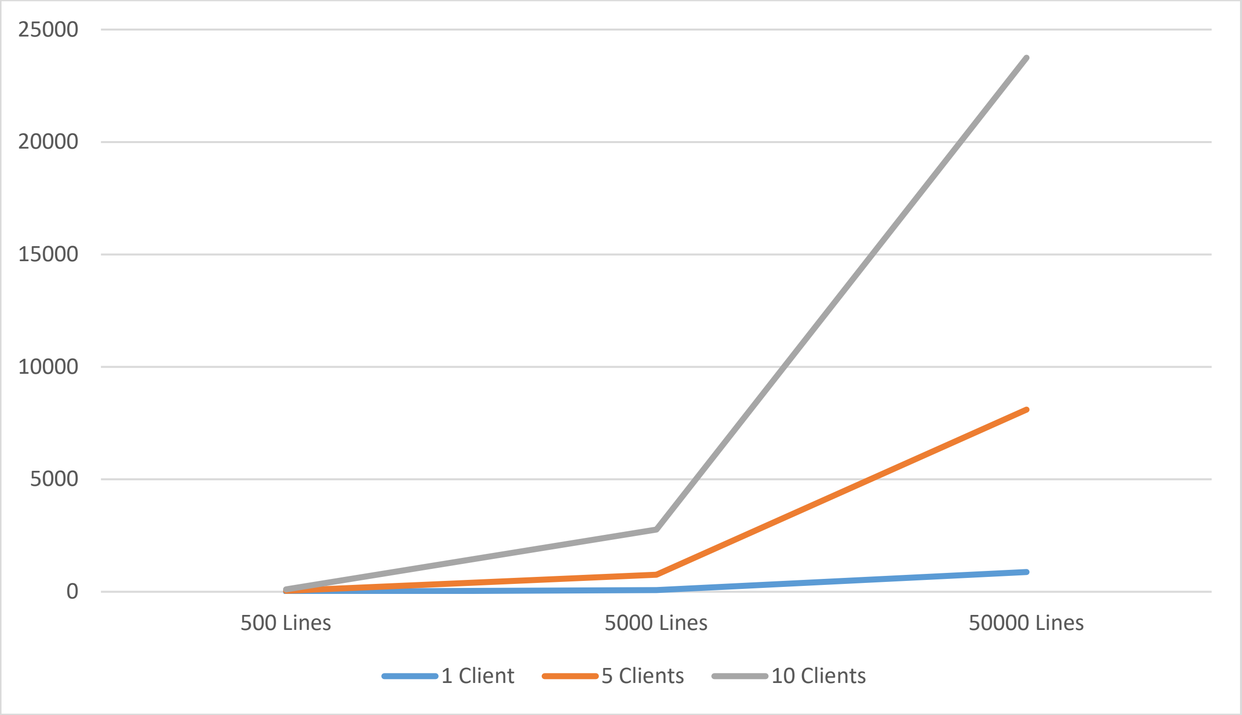
c. 10 Clients

	Number of lines for the input file		
Runs	500	5.000	50.000
1.	104.9	2088.6	23796.0
2.	95.1	2172.3	24404.2
3.	104.8	2001.2	22349.0
4.	104.3	2087.7	24651.6
5.	112.5	2495.5	24286.1
6.	112.1	7864.5	24419.7
7.	121.9	2198.2	23479.6

8.	99.8	2576.3	23798.1
9.	99.9	2144.0	23587.5
10.	101.9	2016.2	22698.4
Average	105.72	2764.45	23747.02

Results and Conclusion

	500 Lines	5000 Lines	50000 Lines
1 Client	0.21	79.48	877.08
5 Clients	41.8	762.54	8099.25
10 Clients	105.72	2764.45	23747.02



The average results in experiments are displayed in the line graph. We can see that increasing thread numbers increases the elapsed time almost linearly. It is because the more clients are created the more thread is supplied for the demand. Our test environment has 1 CPU and all threads take place also the clients and server executes concurrently (not physically). The search algorithm that strstr of c library uses Boyer-Moore Algorithm that have O(m*n) where n = length of string, m = length of pattern. Since pattern length is relatively short, the time required for execution increases linearly. The breakpoint is more precise in more client counts because for each client, one thread is responsible and it causes that context switch lets the process/thread execute very rare. Breakpoint is seen after 5000 lines of input file because in test file, we had identical number of searched keys in each 500 lines. After 5000 lines, we get over the buffer size for result queues, so the thread and client waits for getting and putting values from/to result queue and it increases the execution time.