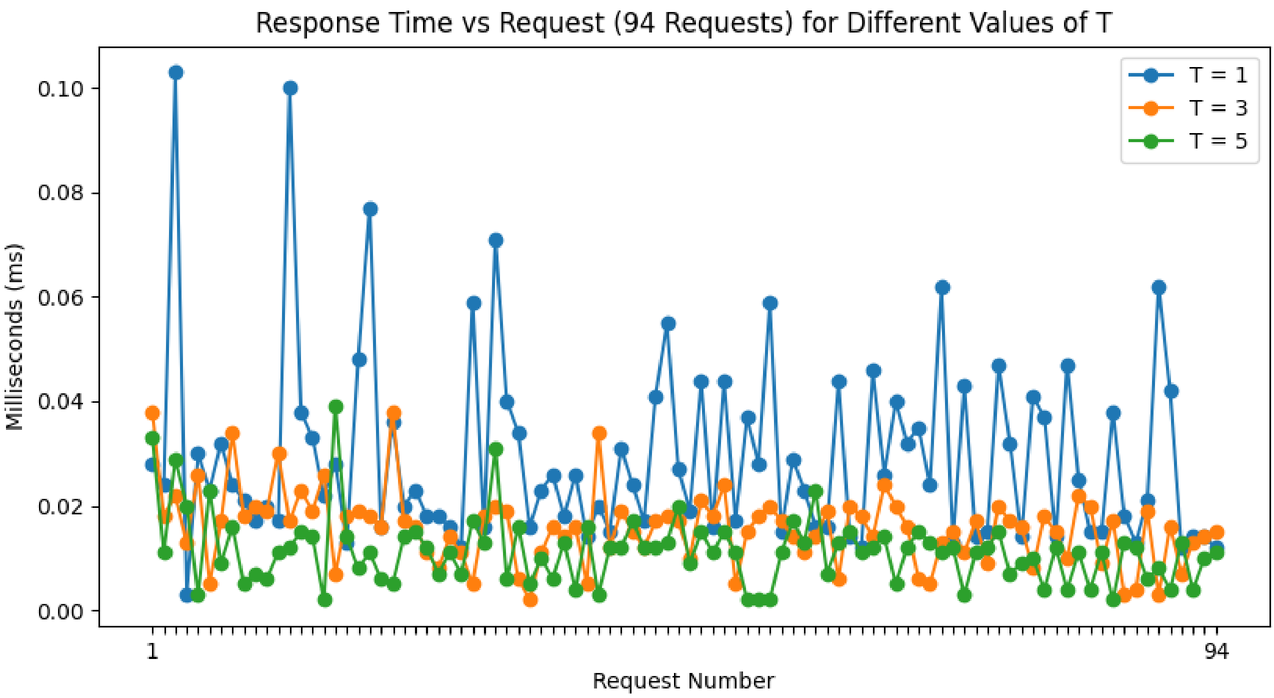
CS 342 - Operating Systems

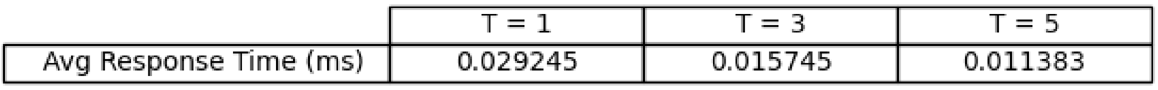
Project 2

Full Name: Arda İynem
ID: 22002717
Section: 1

1. Effect of Worker Thread Amount on Response Time

In this experiment the independent variable is worker thread amount (TCOUNT) which is varied to see the effect on the dependent variable response time. To measure the response time the system clock time is stored with the *clock_gettime(..)* function when a request arrives in the front-end thread of the server program. When the request is assigned to a worker thread, system clock time is stored again. Finally, the difference of two values yields the response time to that specific request. All response time to each 94 requests in input files is measured and then their average is calculated to get average response time. For each different run, D (data file amount) is kept constantly at 5 to not affect the experiment result.



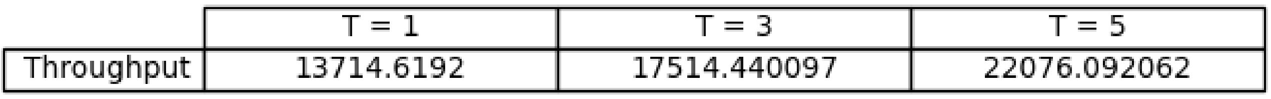


	T = 1	T = 3	T = 5
Avg Response Time (ms)	0.029245	0.015745	0.011383

The result of the experiment shows increasing the worker thread amount decreases the mean response time. This decrease in response time is due to better utilization of parallel programming capacity of the multicore processors. Since the server front thread is able to assign the new requests to idle threads while the others are working on the previous requests when the worker thread amount is increased, the front end thread does not have to wait for a single or a few threads to finish their work before assigning them new work; Thus inevitably the mean response time decreases.

2. Effect of Worker Thread Amount on Throughput

In this experiment the independent variable is worker thread amount (TCOUNT) which is varied to see the effect on the dependent variable throughput. To measure the throughput the system clock time is stored with the *clock_gettime(..)* function when the client threads are created and the first request is sent from a client thread to the server program. When the reply to the last request is assigned to a client thread, system clock time is stored again. Finally, the difference of two values yields the time passed for all 94 requests to be served; By dividing this time value with the request amount (94) I can approximate the throughput. For each different run, D (data file amount) is kept constantly at 5 to not affect the experiment result.

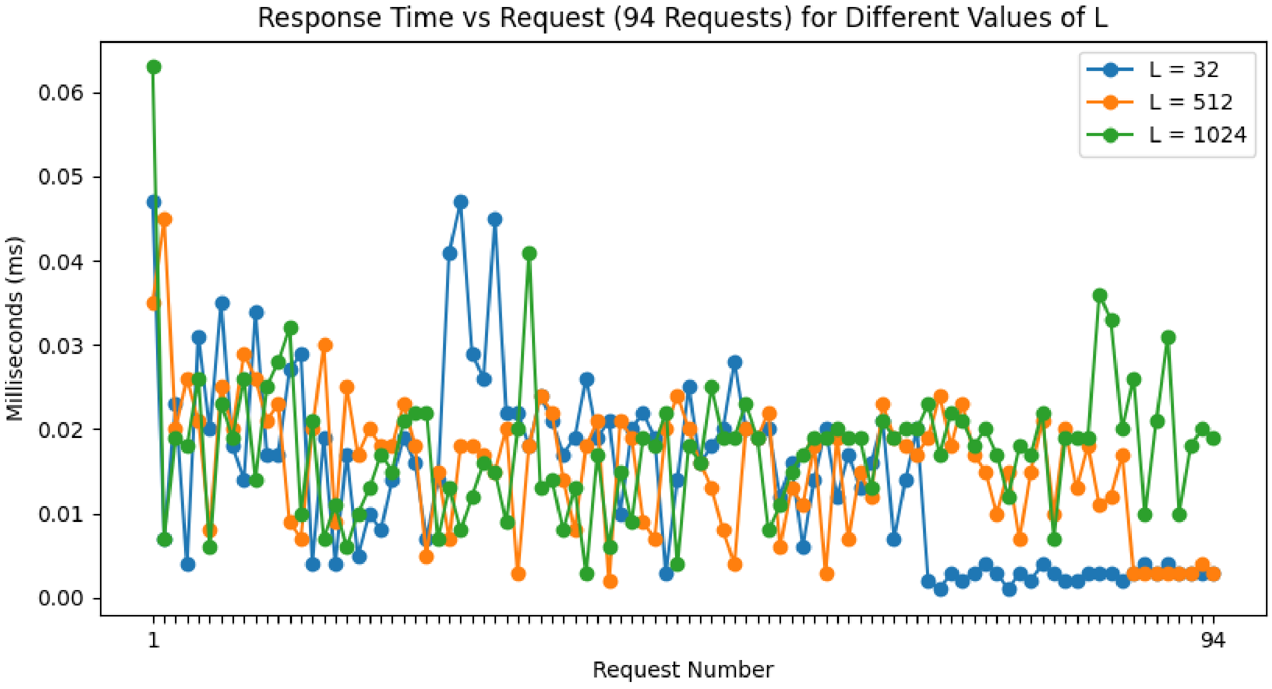


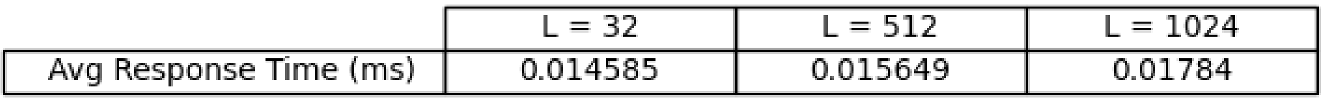
	T = 1	T = 3	T = 5
Throughput	13714.6192	17514.440097	22076.092062

The result of the experiment shows increasing the worker thread amount increases the throughput. This increase in throughput is due to better utilization of parallel programming capacity of the multicore processors. Since the server front thread is able to assign the new requests to idle threads while the others are working on the previous requests when the worker thread amount is increased, the front end thread does not have to wait for a single or a few threads to finish their work before assigning them new work. The simultaneous processing of multiple requests obviously results in more requests to be processed in a time unit. Thus inevitably the throughput increases as worker thread amount increases.

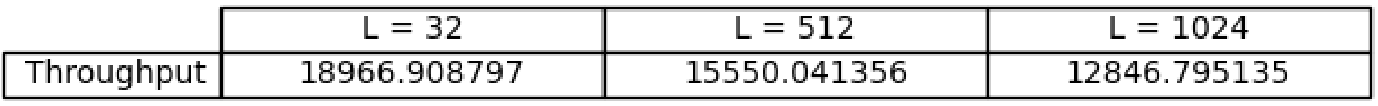
3. Effect of Data Value Size on Response Time & Throughput

In this experiment the independent variable is data value size (VSIZE) which is varied to see the effect on the dependent variable throughput. To measure the response time and throughput the *clock_gettime(..)* function is utilized as aforementioned earlier. For each different run, D (data file amount) is kept constantly at 5 to not affect the experiment result.





	L = 32	L = 512	L = 1024
Avg Response Time (ms)	0.014585	0.015649	0.01784



	L = 32	L = 512	L = 1024
Throughput	18966.908797	15550.041356	12846.795135

The result of the experiment shows increasing the data value size very slightly decreases the mean response time and increases the throughput. This increase in throughput is due to slightly increased processing time of the requests yet this is very subtle and may not be constantly this way in each experiment. Namely, even if the workload of requests increases due to more bytes being read and written; this increase is very insignificant in terms of modern computing power standards. Moreover, mean response times are nearly identical implying that these results may not hold for each run. However, again in theory, the workload of threads slightly increases for each request and they may be ready for new requests just a little bit later when the data value size increases. Even Though this experiment somehow depicted the theory in practice due to limited requests amount (94) and limited increase in data value size (32 to 1024); in real life scenarios with a larger input data set and larger range of data value size, the difference of the response time values and throughput values for different number of data value sizes may become much clear and significant.