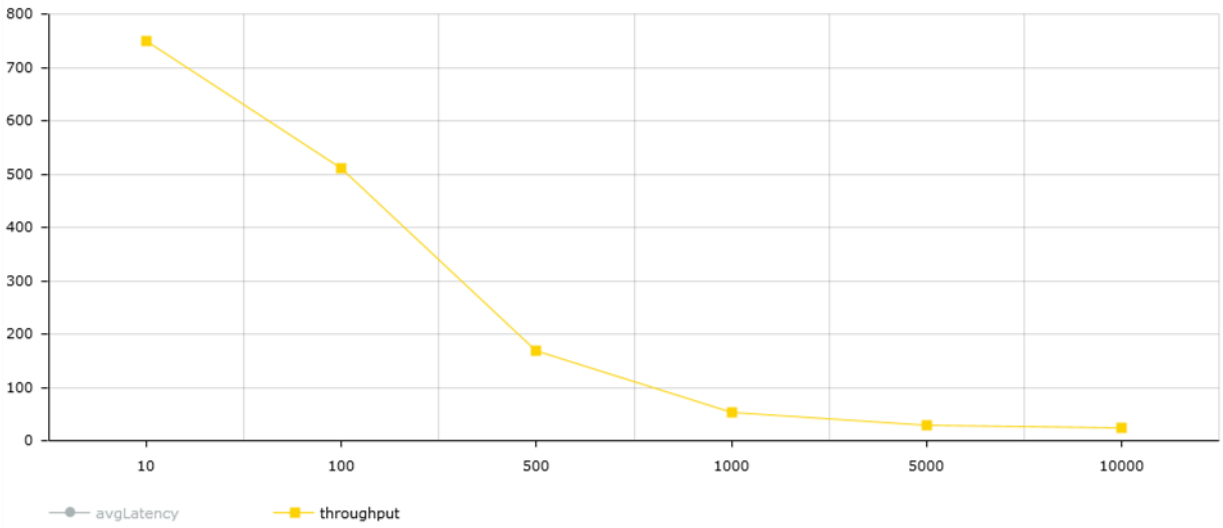
Operating Systems and Concurrency

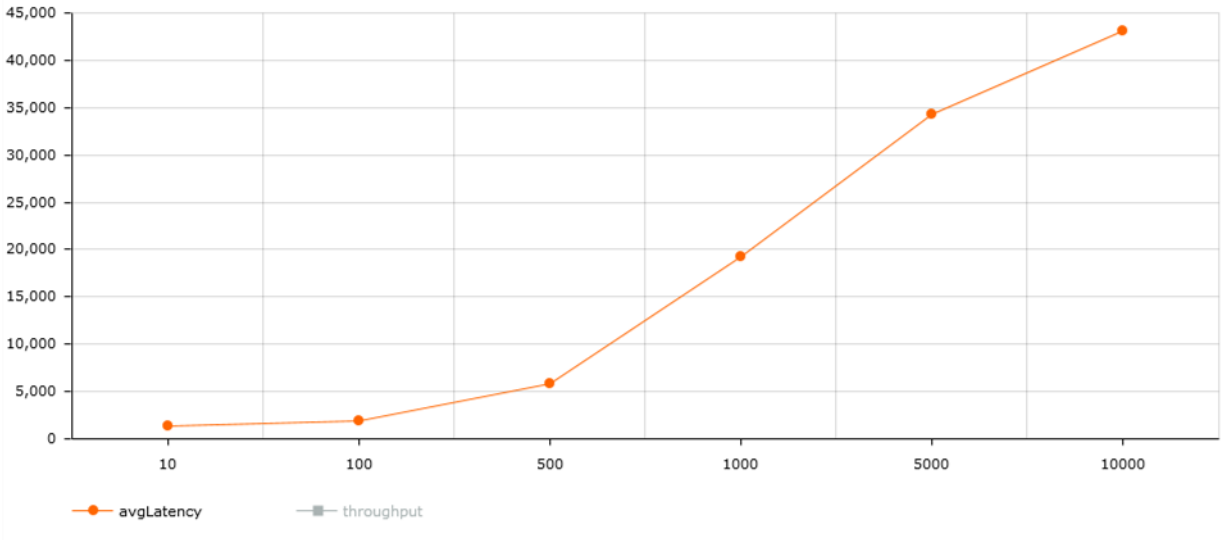
Lab 4 – Graphs and Observations

Part 2

1.Average Latency and Throughput based on N_DATA:

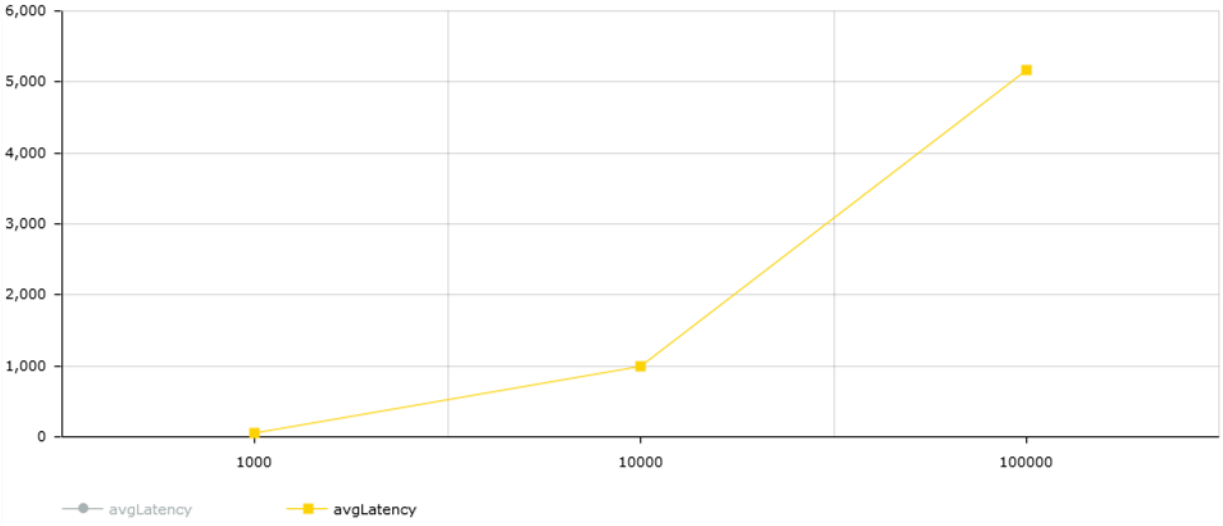
Throughput decreases and average latency increases, the more data we have.

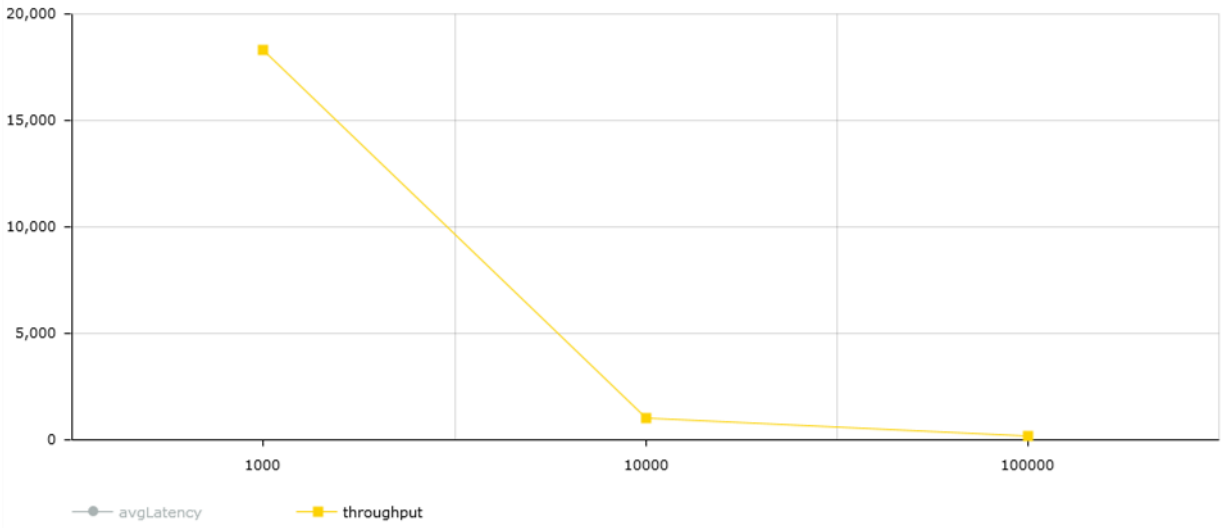




2.Average Latency and Throughput depending on the workload of all threads:

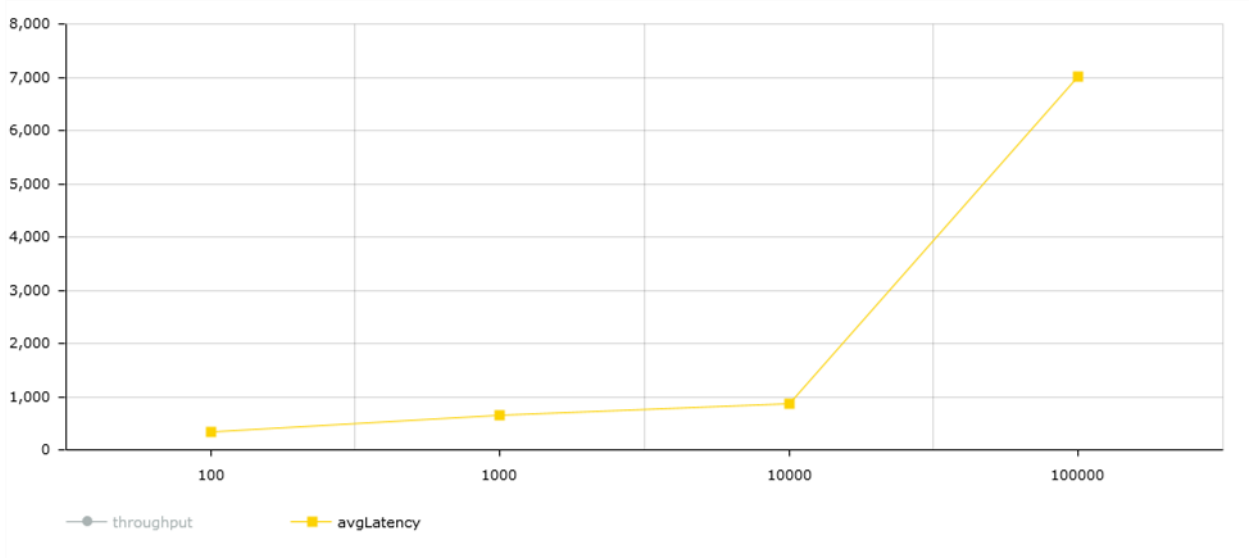
Latency increases and throughput decreases the bigger the workload for the threads is.

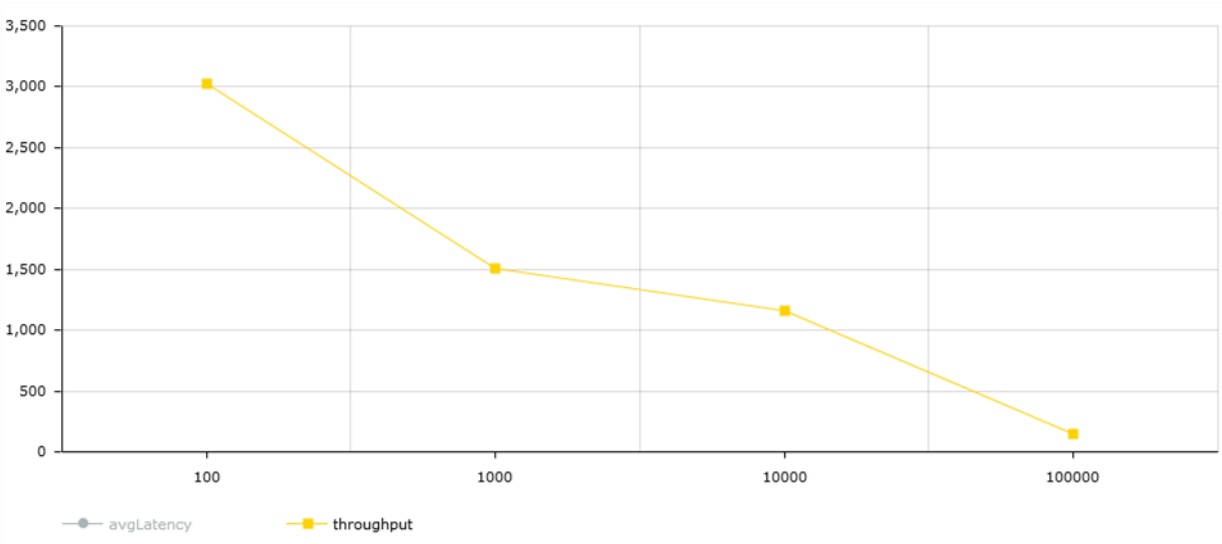




3.Average Latency and Throughput when the first thread has a fixed workload (100.000) and the other 2 threads change workload:

As with the previous test, latency rises and throughput decreases when workload is increased, but the rate at which this happens is lower than before.



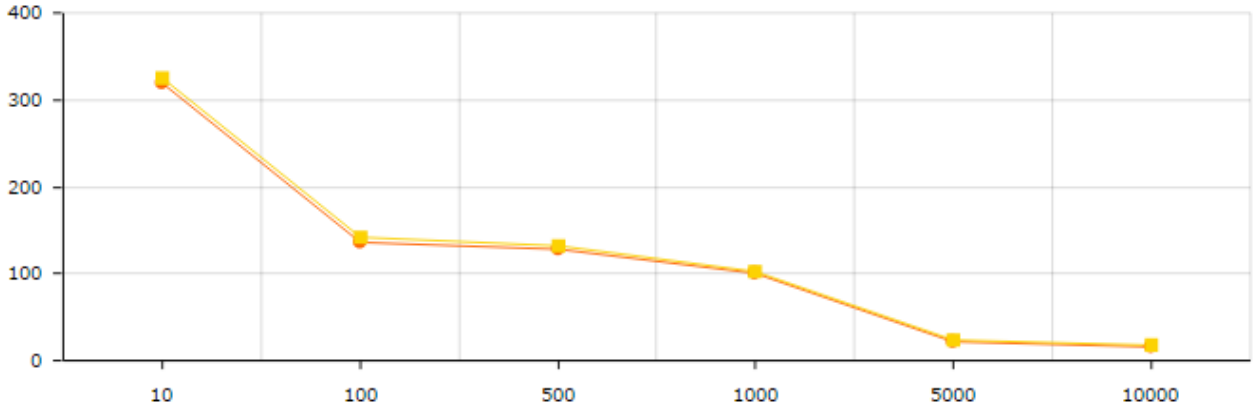


PART 3

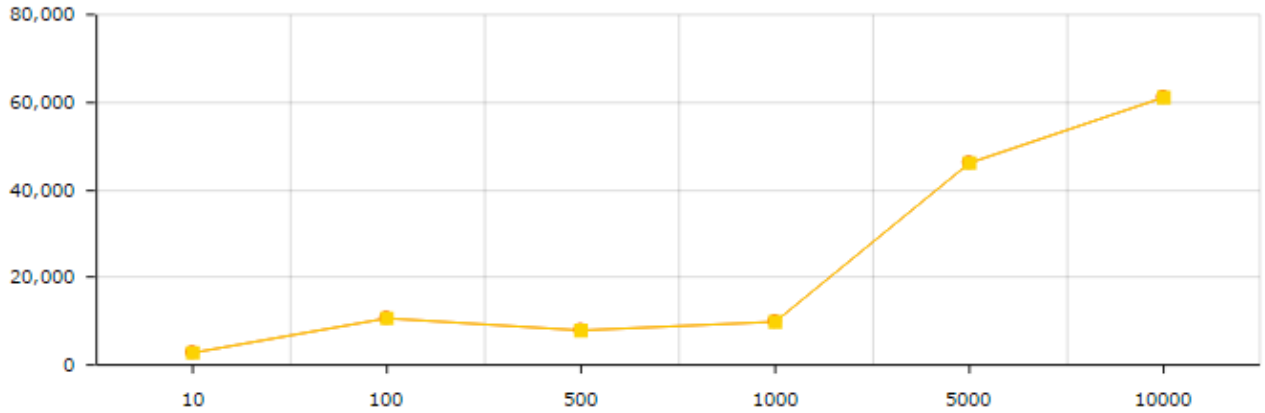
1.Average Latency and Throughput based on N_DATA:

Throughput seems to be dropping, while the average latency is going higher..

Throughput chart

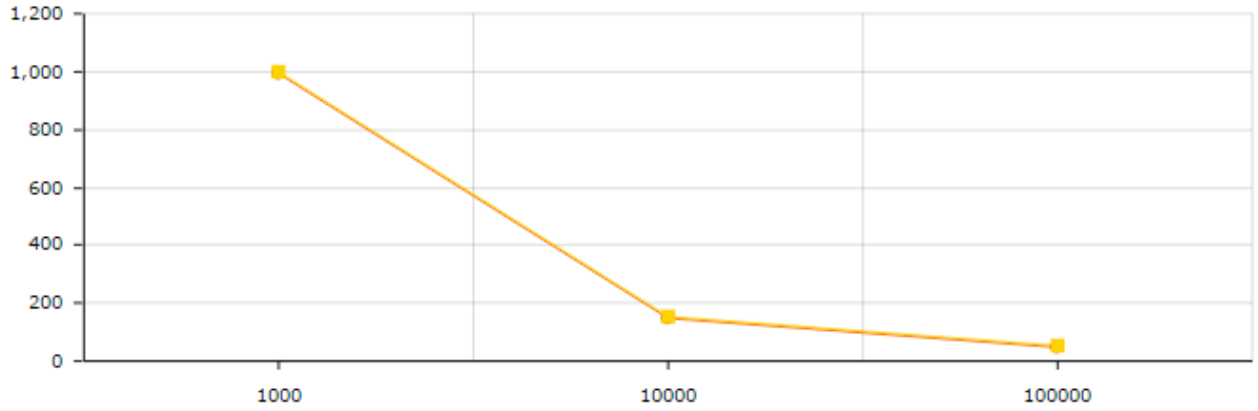


Average Latency chart

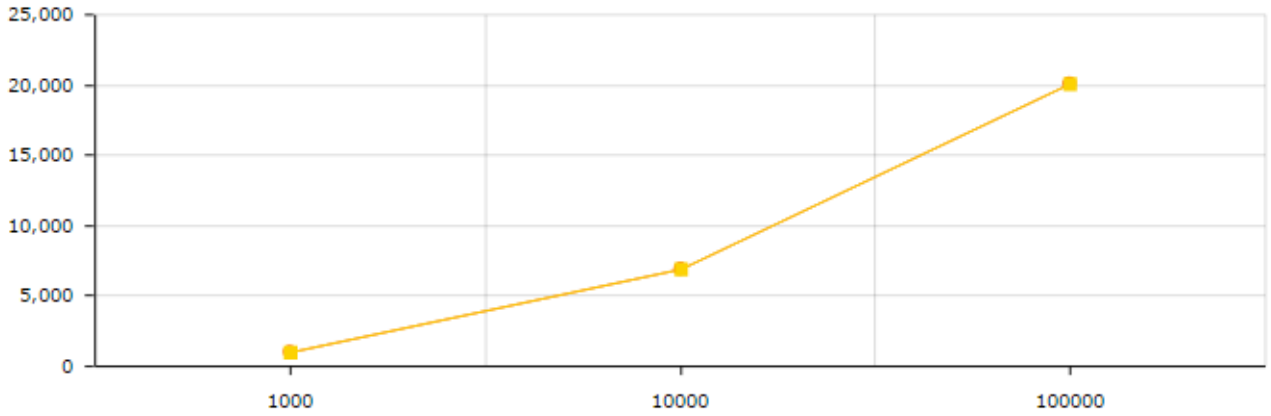


2.Average Latency and Throughput depending on the workload of all threads:

Throughput chart

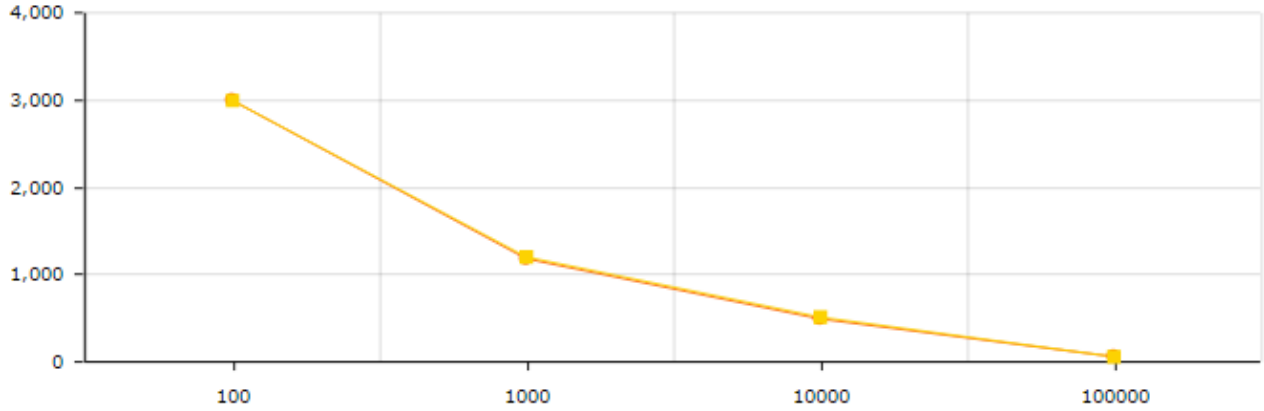


Average Latency Chart

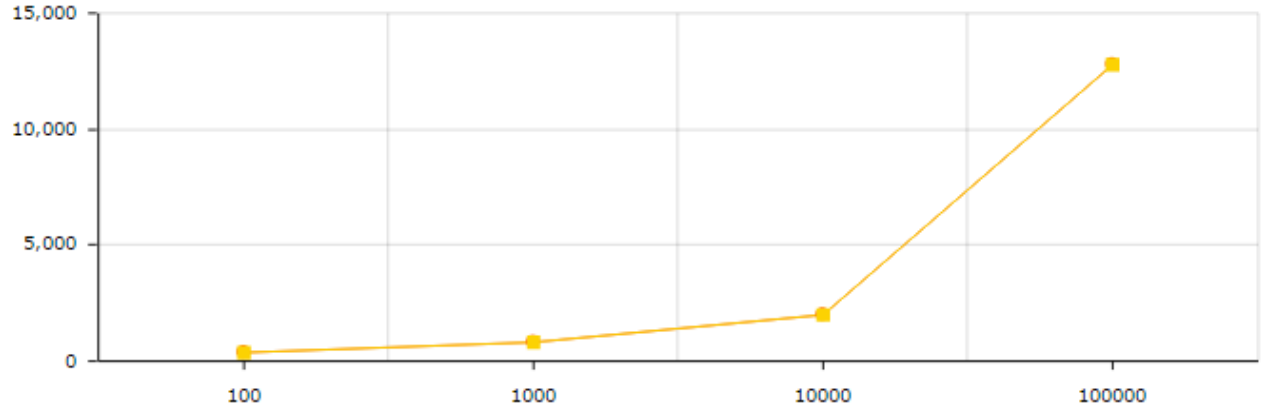


3.Average Latency and Throughput when the first thread has a fixed workload (100.000) and the other 2 threads change workload:

Throughput chart



Average Latency chart

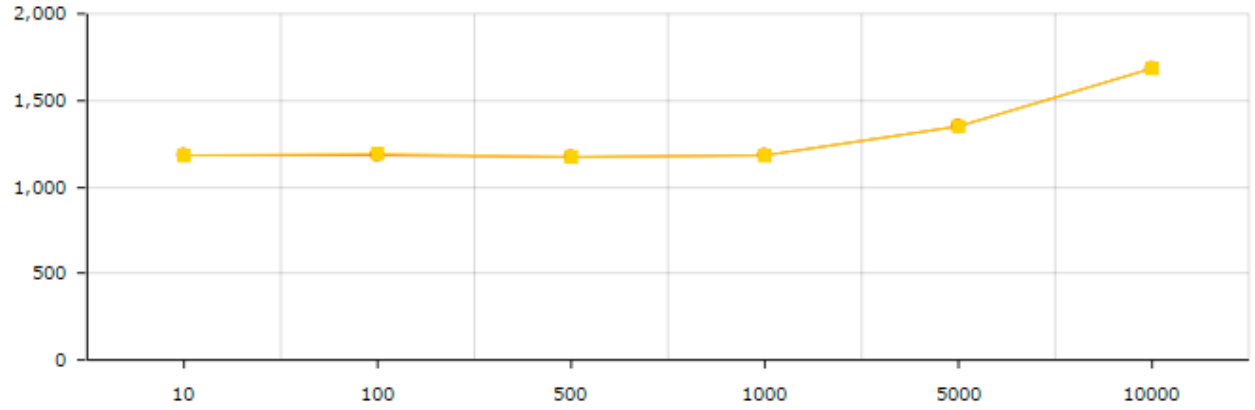


PART 4

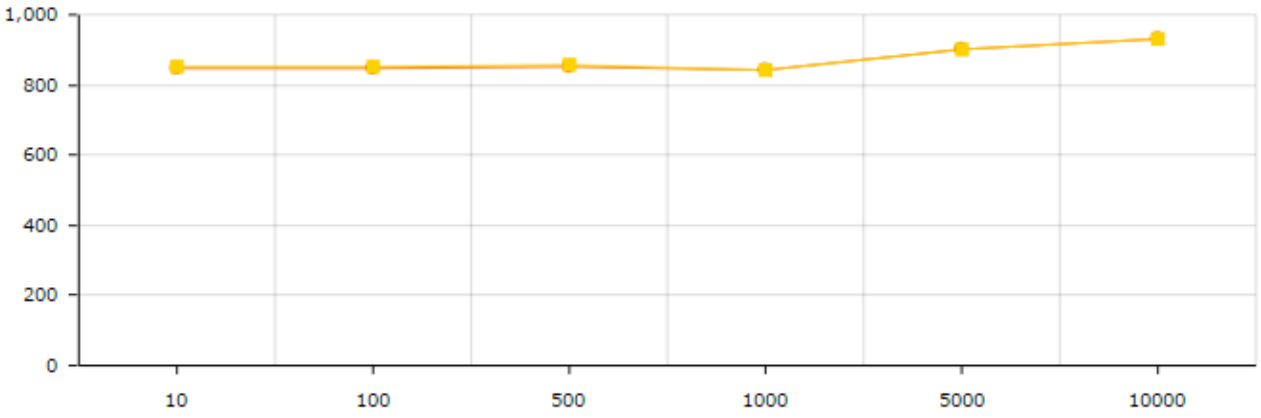
1.Average Latency and Throughput based on N_DATA:

Both latency and throughput seem to be pretty constant in this case. There are some minor increments at times, but I would not register those as dramatic and significant.

Average Latency chart

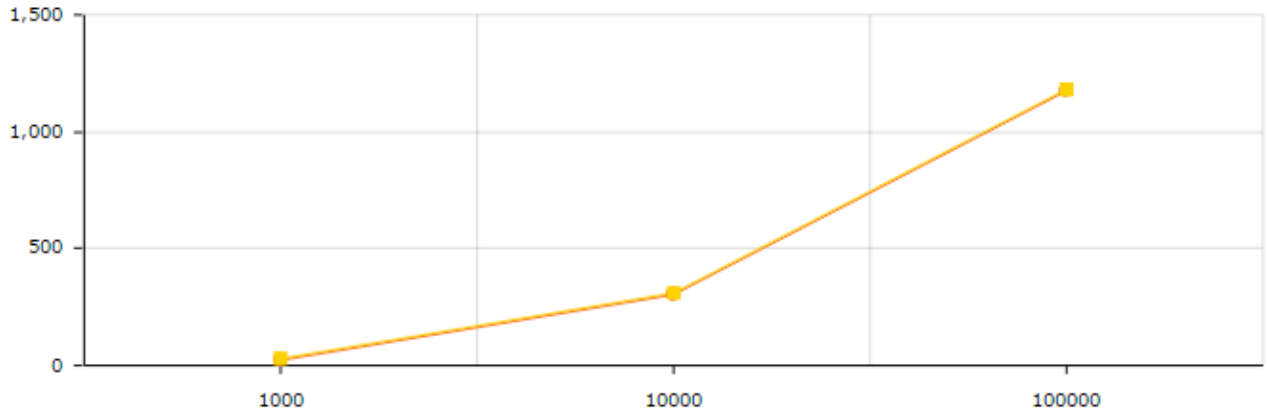


Throughput Chart

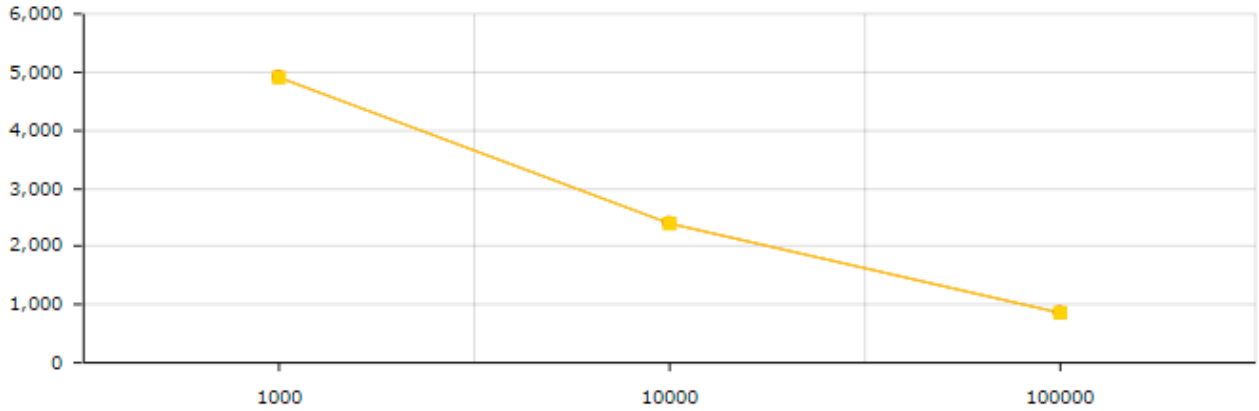


2.Average Latency and Throughput depending on the workload of all threads:

Average Latency Chart

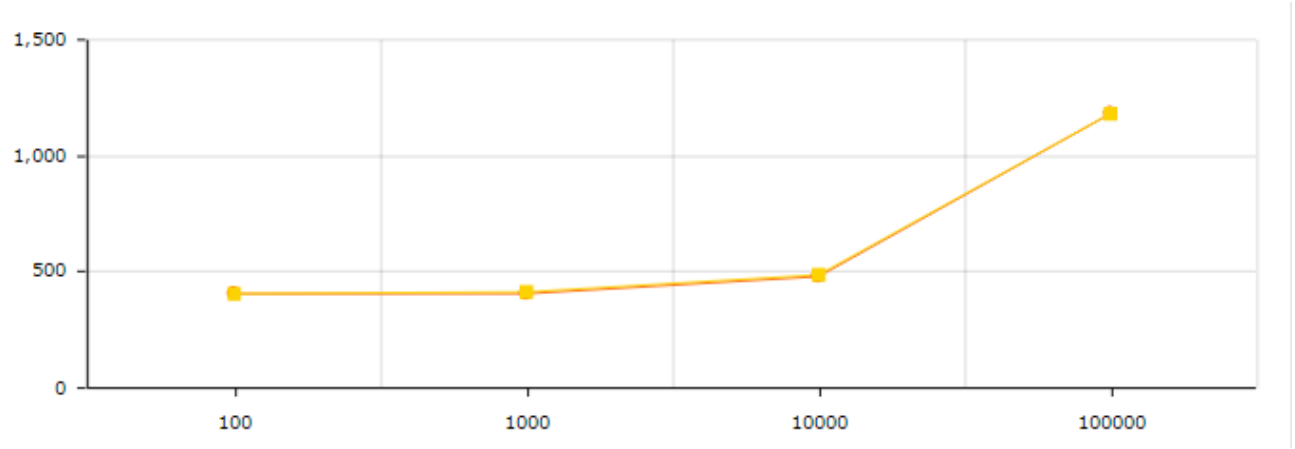


Thread chart

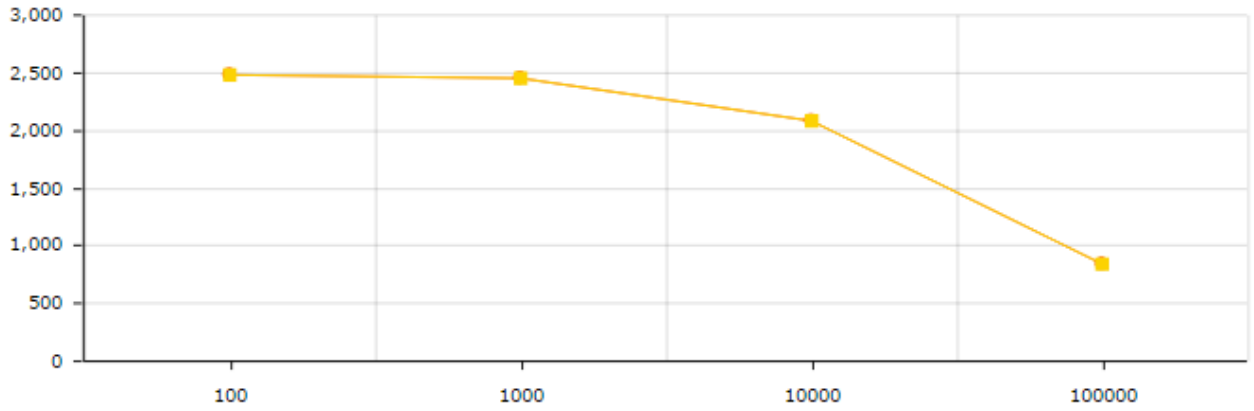


3.Average Latency and Throughput when the first thread has a fixed workload (100.000) and the other 2 threads change workload:

Average Latency Chart



Throughput Chart



I believe the program can be optimised further, by separating the input and output buffers into three areas – one for each thread. We will not need mutexes, for this implementation, because the threads will be operating on their own areas only.