**מאיצים חישוביים ומערכות מואצות**

**046853**

**אביב תשע"ט**

**תרגיל בית 2**

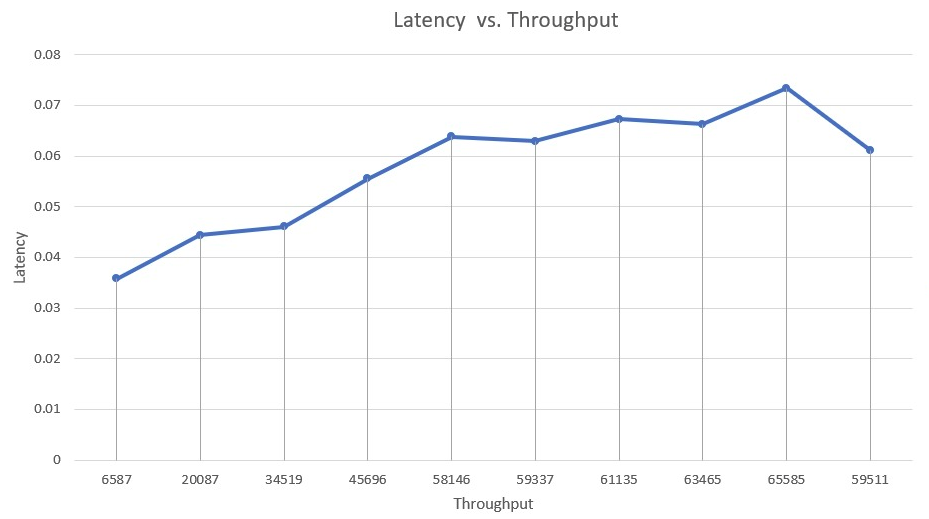
204397368 **:שם:** יחזקאל עידו **תעודת זהות**

305285694 **:שם:** זוהר אוהד **תעודת זהות**

1. CUDA Streams:

Our average max load is: 66,000 [request / sec].

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration # | Load [request/sec] | Throughput [request/sec] | Latency [mSec] |
| 1 | 6,600 | 6587.57 | 0.035831 |
| 2 | 20,533 | 20087.81 | 0.044421 |
| 3 | 34,666 | 34519.61 | 0.046088 |
| 4 | 48,400 | 45696.43 | 0.055533 |
| 5 | 62,333 | 59511.10 | 0.061123 |
| 6 | 76,266 | 58146.55 | 0.063793 |
| 7 | 90,200 | 63465.94 | 0.066275 |
| 8 | 104,133 | 61135.12 | 0.067265 |
| 9 | 118,066 | 59337.43 | 0.062947 |
| 10 | 132,000 | 65585.75 | 0.073375 |

  
We can learn from the graph 3 main conclusion:

* As the throughput is getting higher also the latency is getting higher. We assume this happen because the GPU must handle more request in parallel.
* The throughput is little below the load we set because of some management overheads.
* For some problem (in our case the image Histogram equalization) there is a maximum throughput the GPU can handle using streams, as one can see although the load is higher than 66,000 [request/sec] the throughput never exceed that bound.

1. Producer Consumer Queues:
   1. The way we calculate the maximum number of thread blocks we can invoke is using the method we learned in class:  
      Getting thread block properties:

* Shared memory limit: each of thread blocks is using two arrays of 256 integers, 1 array of 256 bytes and 1-Byte (for request validator) total of: 2305 Bytes.
* Registers limit: using 32 registers.
* Threads per block: depending on user input.

Getting Device properties:

* Shared memory limit: getting shared memory per SM.
* Registers limit: using getting registers per SM.
* Threads per block: max number of threads per SM.

After getting all the properties we calculate the number of thread blocks per SM by each limit (Hardware limitation / Block requirements)and the minimum was chosen.