**CS575:Parallel Programming**
**OpenCL Array Multiply, Multiply-Add, and Multiply-Reduce**

**Student Name: Ping-Jui Liao**
**Student Email: liaop@oregonstate.edu**
**Project Number: 5**
**Project Name: OpenCL Array Multiply, Multiply-Add, and Multiply-Reduce**

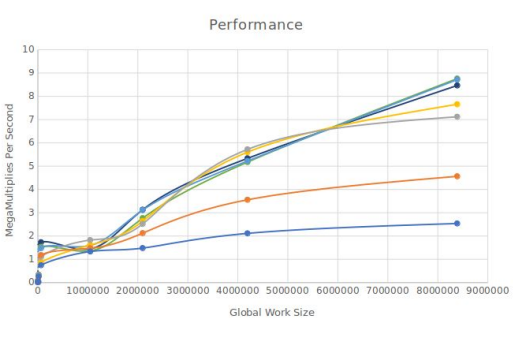
1. **Show this table and graph**

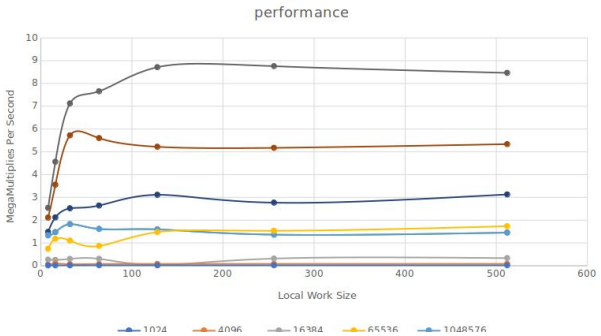
**OpenCL Array Multiply**

| run-1 | 2 to the 10 | 2 to the 12 | 2 to the 14 | 2 to the 16 | 2 to the 20 | 2 to the 22 | 2 to the 24 | 2 to the 25 |
|---|---|---|---|---|---|---|---|---|
| | 1024 | 4096 | 16384 | 65536 | 1048576 | 2097152 | 4194304 | 8388608 |
| 8 | 0.014 | 0.04 | 0.266 | 0.745 | 1.333 | 1.481 | 2.112 | 2.537 |
| 16 | 0.015 | 0.094 | 0.255 | 1.179 | 1.474 | 2.121 | 3.555 | 4.562 |
| 32 | 0.016 | 0.068 | 0.298 | 1.102 | 1.824 | 2.518 | 5.722 | 7.121 |
| 64 | 0.017 | 0.074 | 0.302 | 0.866 | 1.614 | 2.641 | 5.601 | 7.657 |
| 128 | 0.017 | 0.08 | 0.0311 | 1.474 | 1.595 | 3.112 | 5.221 | 8.714 |
| 256 | 0.017 | 0.08 | 0.313 | 1.532 | 1.363 | 2.767 | 5.174 | 8.759 |
| 512 | 0.017 | 0.082 | 0.333 | 1.731 | 1.452 | 3.128 | 5.335 | 8.465 |

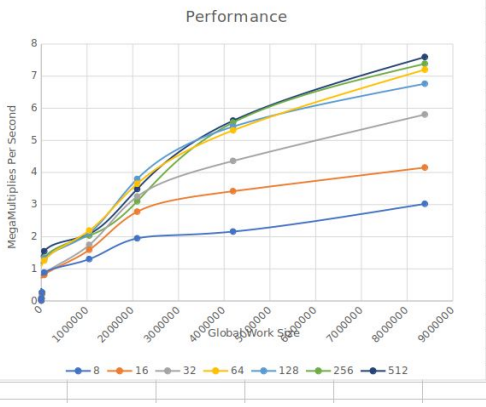
**OpenCL Array Multiply-add**

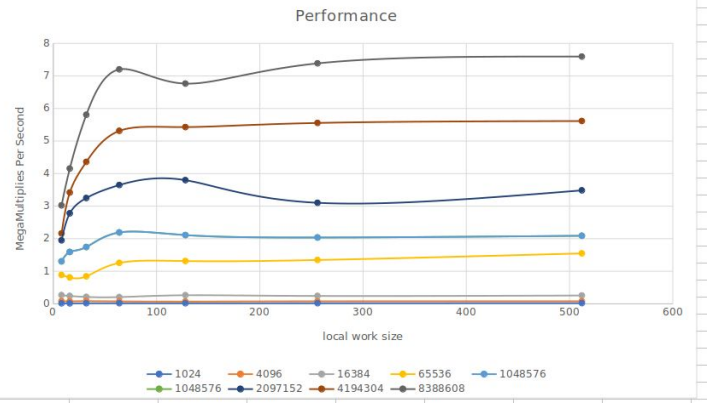
`



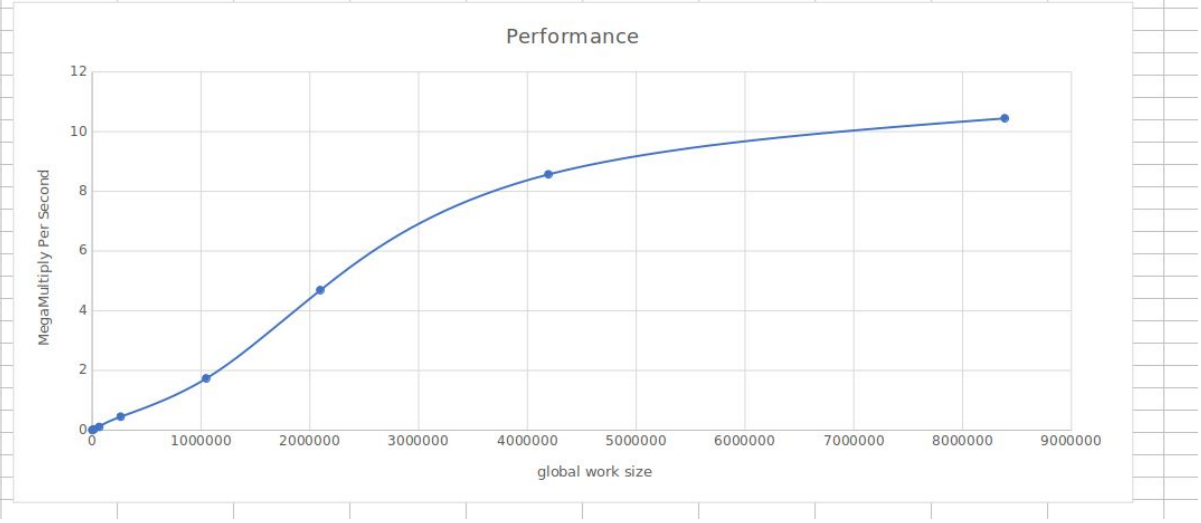

| run-2 | 2 to the 10 | 2 to the 12 | 2 to the 14 | 2 to the 16 | 2 to the 20 | 2 to the 22 | 2 to the 24 | 2 to the 25 |
|---|---|---|---|---|---|---|---|---|
| | 1024 | 4096 | 16384 | 65536 | 1048576 | 2097152 | 4194304 | 8388608 |
| 8 | 0.013 | 0.081 | 0.265 | 0.885 | 1.303 | 1.949 | 2.16 | 3.023 |
| 16 | 0.012 | 0.076 | 0.241 | 0.81 | 1.592 | 2.778 | 3.417 | 4.153 |
| 32 | 0.016 | 0.082 | 0.211 | 0.842 | 1.744 | 3.249 | 4.359 | 5.805 |
| 64 | 0.017 | 0.075 | 0.206 | 1.256 | 2.191 | 3.648 | 5.312 | 7.202 |
| 128 | 0.016 | 0.069 | 0.263 | 1.313 | 2.111 | 3.798 | 5.428 | 6.764 |
| 256 | 0.017 | 0.077 | 0.241 | 1.345 | 2.036 | 3.102 | 5.555 | 7.386 |
| 512 | 0.02 | 0.078 | 0.255 | 1.548 | 2.089 | 3.484 | 5.614 | 7.595 |




## OpenCL Array Multiply-reduce

| run-3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1024 | 4096 | 16384 | 65536 | 262144 | 1048576 | 2097152 | 4194304 | 8388608 |
| 32 | 0.00167 | 0.007 | 0.029 | 0.108 | 0.45 | 1.732 | 4.688 | 8.567 | 10.444 |




2. **What pattern are you seeing in this performance curve?**

   Most of the growth are linear, except some anomaly in the graph.

   I believe giving enough times of run would make the graph smoother.

   However, I think small chunk of work size would definitely result in  weird growth because the parallelism does not have significant impact on performance, while cache miss and other system factors are more influential

3. **Why do you think the pattern looks this way?**

   Apparently,  parallelism grows with local work size and global work size. However, some anomalies exist. I believe that these anomalies come with system wide factor. I am using my ubuntu 18.04 machine in my laptop. I believe laptop is not designed for parallelism and the data can be processed are limited.

4. **What does that mean for the proper use of GPU parallel computing?**

   GPU parallel computing is solid. It’s really good for parallelism. The performance is better than just using openmp for spawning threads.