Project #5: OpenCL Array Multiply, Multiply-Add, and Multiply-Reduce

Danlin Song

songdan@oregonstate.edu

1. What machine you ran this on
   I ran this project on OSU rabbit server.
2. Show the tables and graphs

```
#define NUM_ELEMENTS NMB * 1024
```

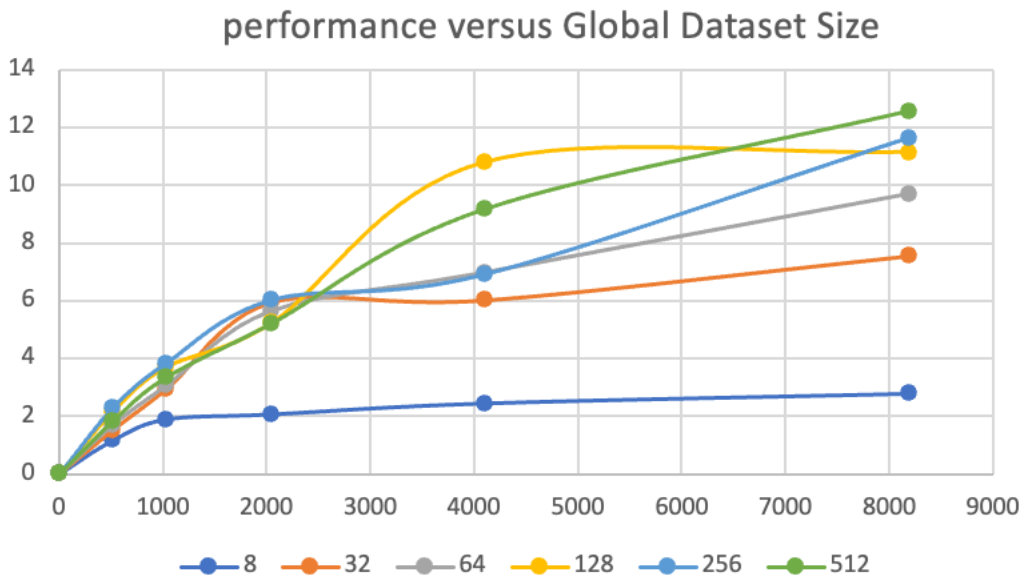
- Multiply table

	8	32	64	128	256	512
1	0.012	0.01	0.016	0.011	0.012	0.01
512	1.166	1.528	1.709	2.166	2.295	1.834
1024	1.891	2.938	3.061	3.701	3.834	3.342
2048	2.069	5.96	5.654	5.253	6.035	5.231
4096	2.445	6.028	6.978	10.802	6.92	9.174
8192	2.783	7.562	9.698	11.152	11.659	12.577

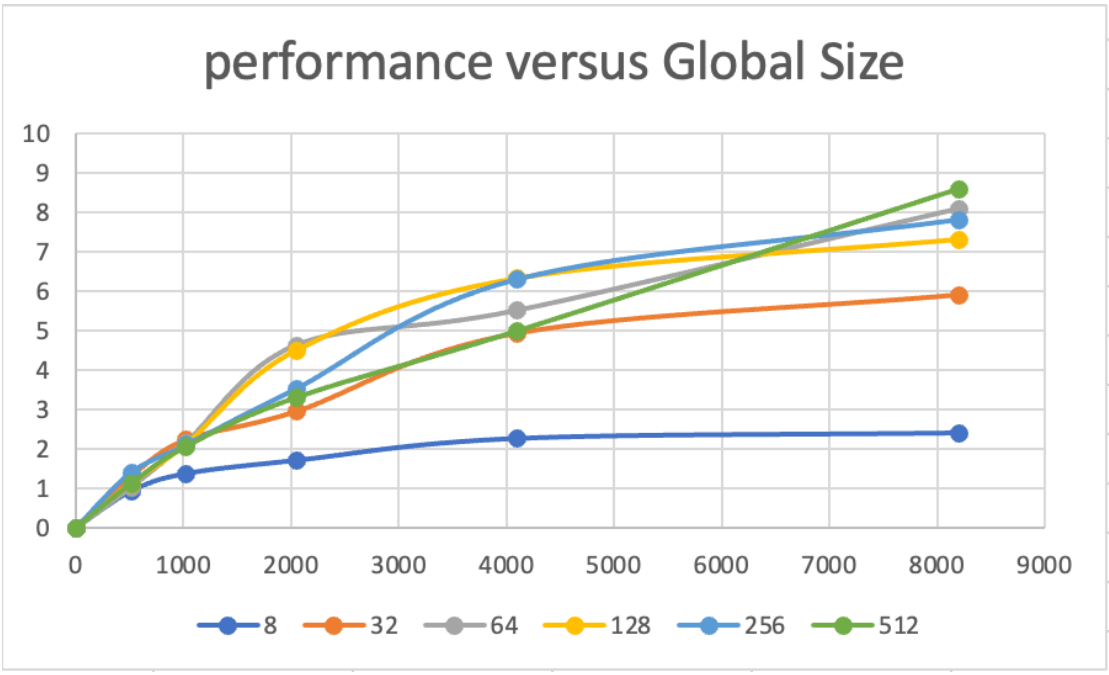
- Multiply-Add table

	8	32	64	128	256	512
1	0.009	0.014	0.011	0.01	0.017	0.011
512	0.941	1.313	1.036	1.14	1.413	1.127
1024	1.388	2.248	2.181	2.134	2.124	2.072
2048	1.724	2.967	4.649	4.518	3.549	3.314
4096	2.281	4.952	5.529	6.347	6.323	4.999
8192	2.422	5.924	8.11	7.336	7.836	8.598

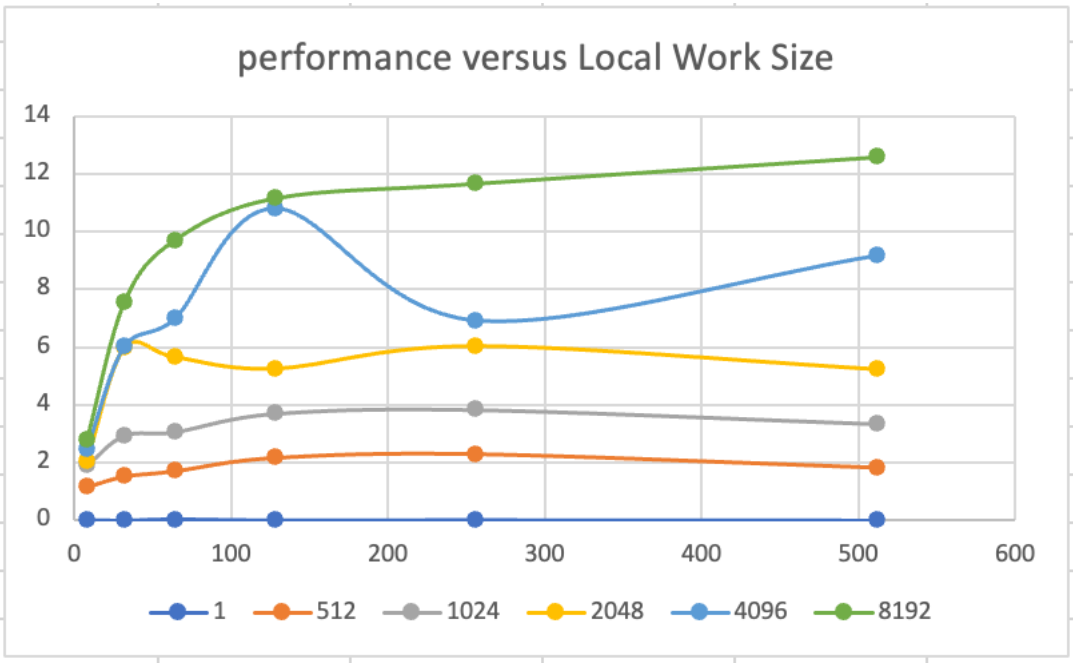
- Multiply and Multiply-Add performance versus Global Dataset Size, with a series of colored Constant-Local-Work-Size curves
  - Multiply



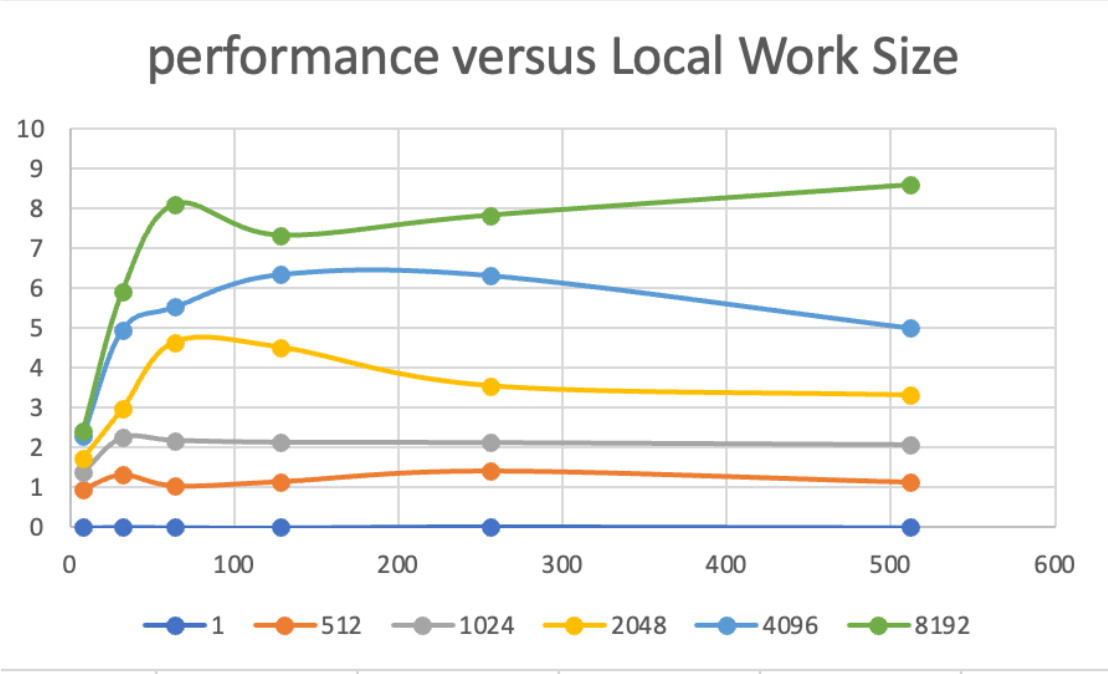
  - Multiply-Add



- Multiply and Multiply-Add performance versus Local Work Size, with a series of colored Constant-Global-Dataset-Size curves
    - Multiply



- Multiply-Add



3. What patterns are you seeing in the performance curves?
   Better performance is obtained with increasing global datasets and local work sizes. However, the increase slows down when the global datasets increase to a certain point, and in the case of local work size, the increase even decreases to a certain point. In general, however, more data generally results in better performance.
4. Why do you think the patterns look this way?
   When the size of the workgroup is large enough, the program will calculate the size of that large array in every second. However, when the size of the workgroup exceeds what the local machine can carry, the rate of performance increase decreases.
5. What is the performance difference between doing a Multiply and doing a Multiply-Add?
   Because there is an extra addition operation, the performance of multiply-add is relatively degraded. Moreover, this performance is unstable because the presence of the addition step can cause some problems in multi-threaded operations.
6. What does that mean for the proper use of GPU parallel computing?
   Check the local work size before running to determine the proper size and try to keep the operation formula as simple as possible; too many operating methods in the formula may lead to performance degradation.

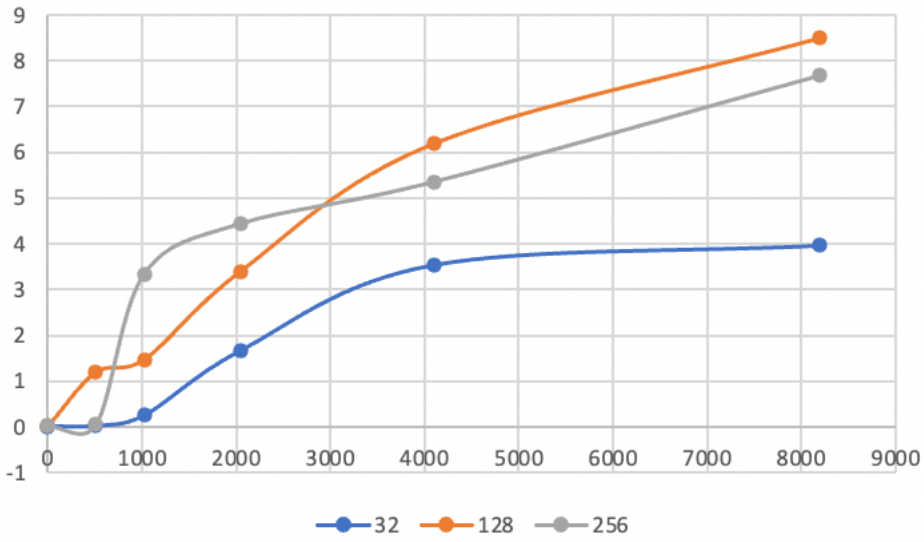
Multiply-Reductions

```
#define NUM_ELEMENTS NMB * 1024
```

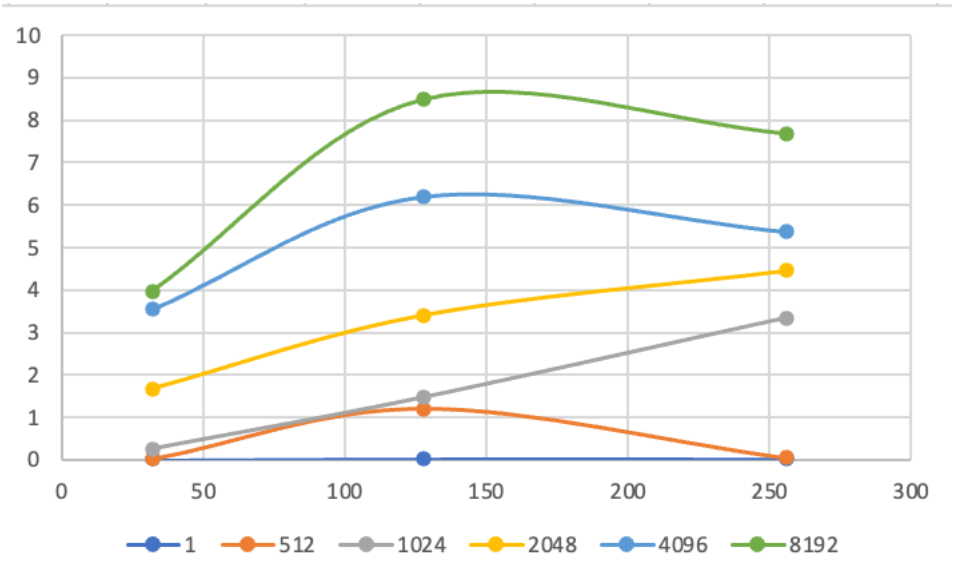
1. Show this table and graph

	32	128	256
1	0.013	0.018	0.018
512	0.023	1.205	0.047
1024	0.252	1.472	3.346
2048	1.667	3.403	4.45
4096	3.536	6.19	5.362
8192	3.962	8.495	7.685

Constant work size:



Constant datasize:



2. What pattern are you seeing in this performance curve?
   The performance of multiplicative reduction scales reliably, but the size of the workgroups has a greater impact on performance growth. Performance continues to grow beyond a workgroup size of 128. It continues to grow to at least 256 workgroups and is likely to continue to grow.
3. Why do you think the pattern looks this way?
   As the array size increases, so does the performance. Most of the work is done on the GPU array, so the remaining part that needs to be done serially on the CPU is the final summation of the values in the returned workgroup. This allows the GPU to process many values in parallel, and then the CPU only needs to do a smaller final summation of the following data values in the array, which come from a pipeline in the cache
4. What does that mean for the proper use of GPU parallel computing?
   Match the GPU with values that are appropriate for the data size and work size; too many bits can significantly degrade performance. Initial testing of the OpenCL program should be performed to evaluate system-specific boundaries before determining final values for group size and dataset size. In addition, this proof of concept can be done with a CPU-only version to demonstrate the value of both solutions before advancing to the final deployed solution.