Long Mach

machl@oregonstate.edu

CS 475- 400

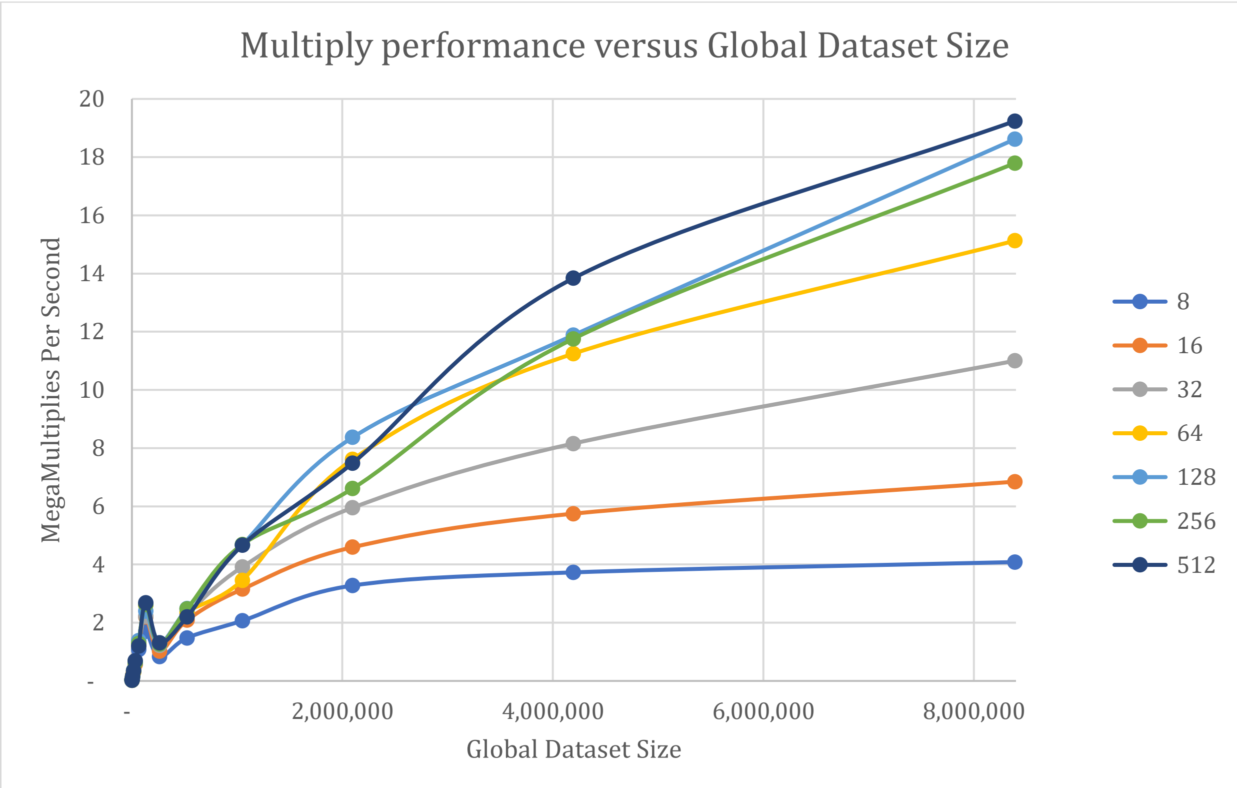
Project #6- OpenCL Array Multiply, Multiply-Add, and Multiply-Reduce

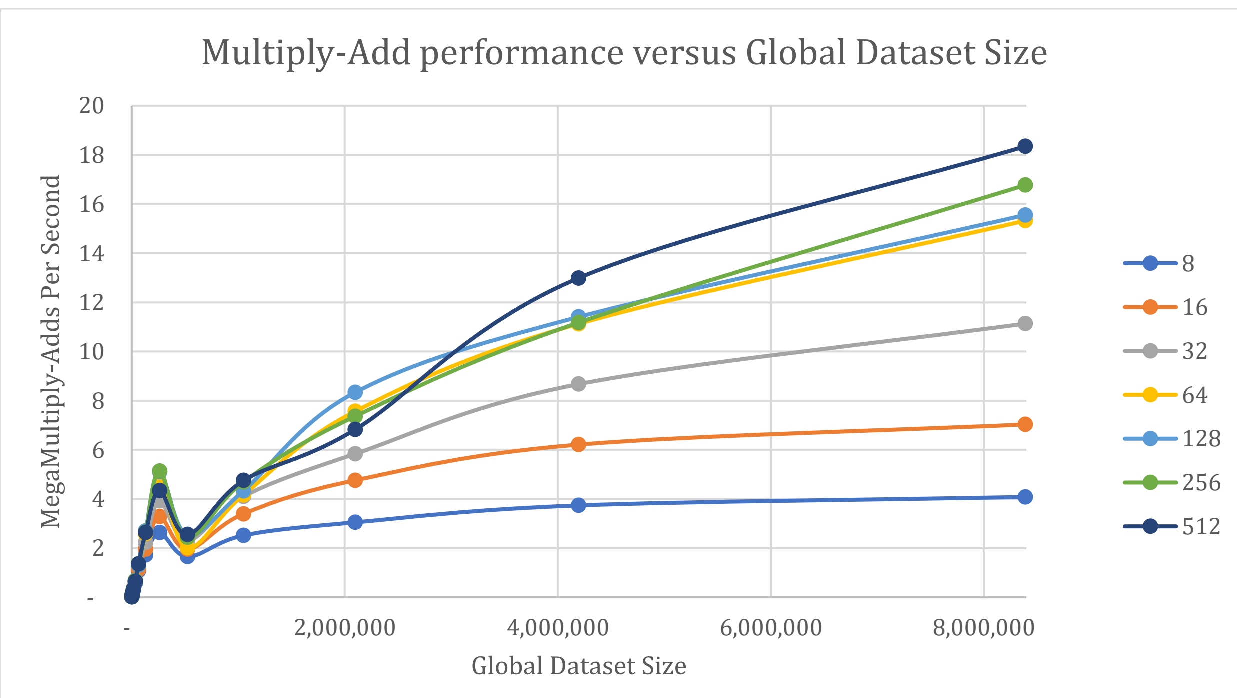
1- First and Second Parts:

Multiply performance result	Local Work Size						
Global Dataset Size	8	16	32	64	128	256	512
1,024	0.021	0.023	0.021	0.023	0.022	0.021	0.021
2,048	0.039	0.040	0.038	0.044	0.045	0.043	0.040
4,096	0.087	0.086	0.090	0.090	0.081	0.091	0.080
8,192	0.176	0.175	0.153	0.176	0.156	0.167	0.159
16,384	0.316	0.334	0.310	0.310	0.357	0.356	0.355
32,768	0.597	0.575	0.663	0.674	0.644	0.632	0.689
65,536	1.080	1.232	1.285	1.284	1.387	1.290	1.203
131,072	1.710	2.201	2.227	2.608	2.395	2.629	2.681
262,144	0.831	1.027	1.217	1.292	1.287	1.265	1.307
524,288	1.478	2.087	2.283	2.364	2.483	2.475	2.200
1,048,576	2.068	3.152	3.914	3.459	4.671	4.685	4.661
2,097,152	3.278	4.594	5.953	7.613	8.369	6.611	7.486
4,194,304	3.730	5.748	8.157	11.246	11.880	11.756	13.837
8,388,608	4.084	6.846	11.000	15.125	18.614	17.789	19.238

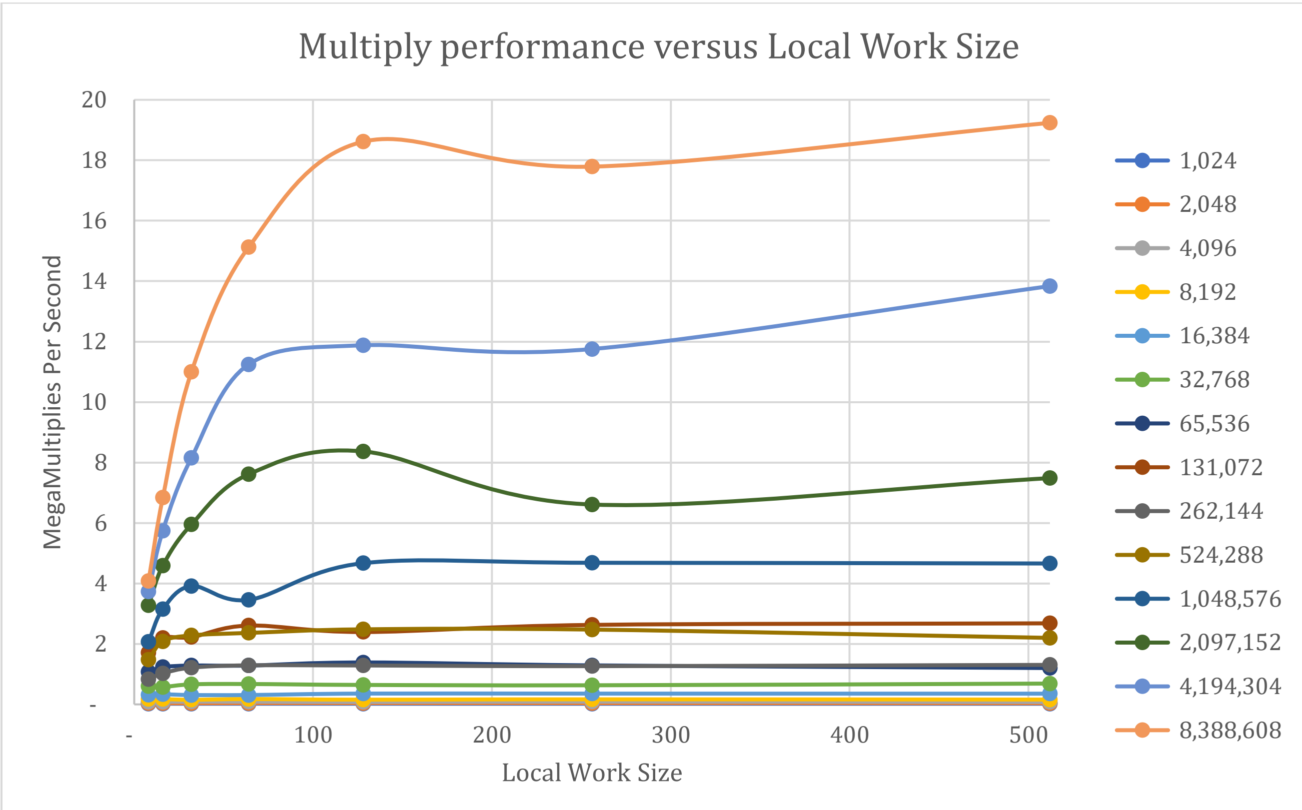
Multiply-Add performance result	Local Work Size						
Global Dataset Size	8	16	32	64	128	256	512
1,024	0.019	0.021	0.021	0.022	0.021	0.019	0.023
2,048	0.041	0.044	0.037	0.044	0.039	0.036	0.042
4,096	0.085	0.084	0.084	0.084	0.083	0.088	0.067
8,192	0.161	0.174	0.153	0.154	0.175	0.176	0.179
16,384	0.308	0.344	0.342	0.347	0.333	0.313	0.327
32,768	0.579	0.648	0.640	0.677	0.685	0.674	0.637
65,536	1.085	1.138	1.302	1.350	1.303	1.355	1.365
131,072	1.724	1.954	2.225	2.585	2.700	2.646	2.634
262,144	2.628	3.288	3.919	4.658	5.113	5.139	4.341
524,288	1.655	1.956	2.307	2.012	2.531	2.442	2.554
1,048,576	2.517	3.394	4.104	4.161	4.329	4.693	4.750
2,097,152	3.050	4.760	5.836	7.563	8.335	7.364	6.823
4,194,304	3.739	6.212	8.671	11.125	11.404	11.175	12.985
8,388,608	4.081	7.033	11.134	15.321	15.553	16.769	18.344

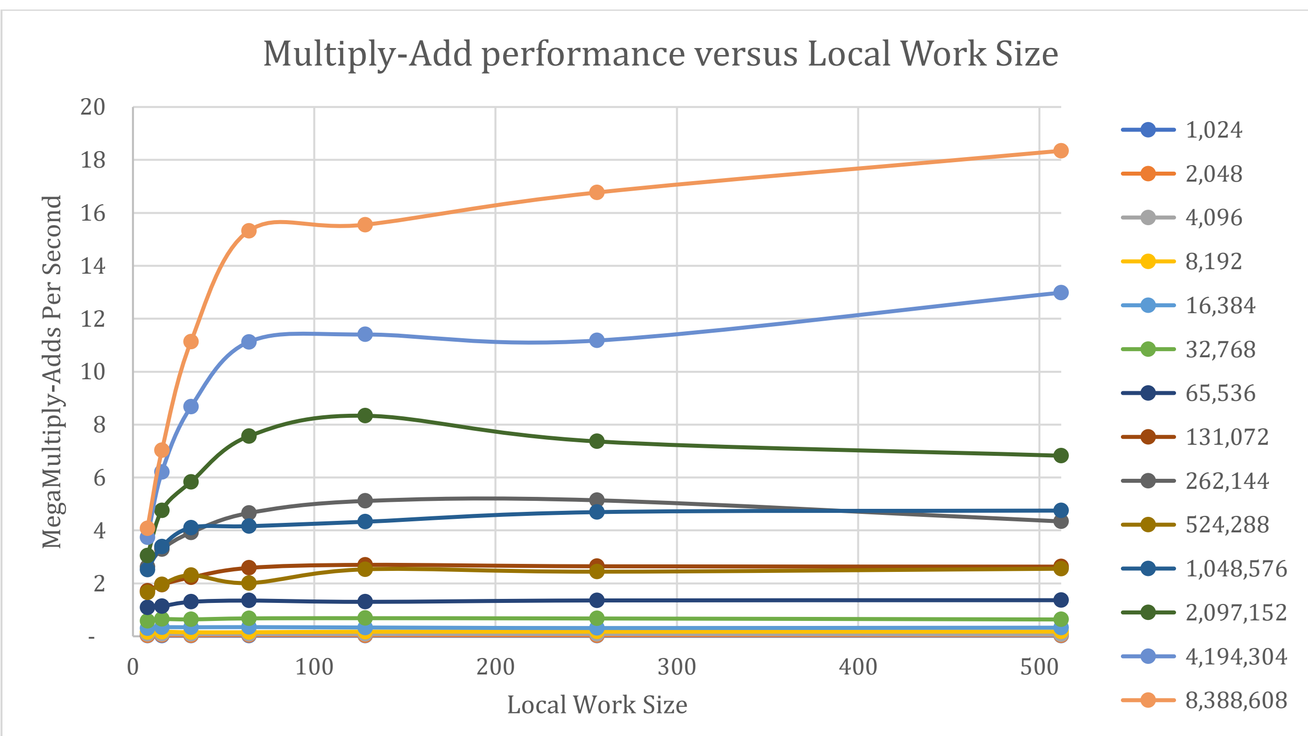
a- Multiply and Multiply-Add performance versus Global Dataset Size, with a series of colored Constant-Local-Work-Size curves





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





2- First and Second Parts Commentary:

a) What machine you ran this on:

   I ran this on DGX machine.

b) What patterns are you seeing in the performance curves?

   In both parts, the larger the Local Work Size or the larger the Global Dataset Size, the higher the performance.

c) Why do you think the patterns look this way?

   For larger Local Work Size, each processing element can share memory and synchronize with other threads in the same work group so it will increase the performance.

   For larger Global Dataset Size, it can utilize the GPU parallel computing benefits from OpenCL. If the Global Dataset Size is too small, the overhead cost of parallel computing will cancel the benefit.

d) What is the performance difference between doing a Multiply and doing a Multiply-Add?

   There seems to be little to no difference between doing a Multiply and doing a Multiply-Add. It means that the Multiply-add was using FMA instruction instead of two separated multiply and adding instructions.

e) What does that mean for the proper use of GPU parallel computing?

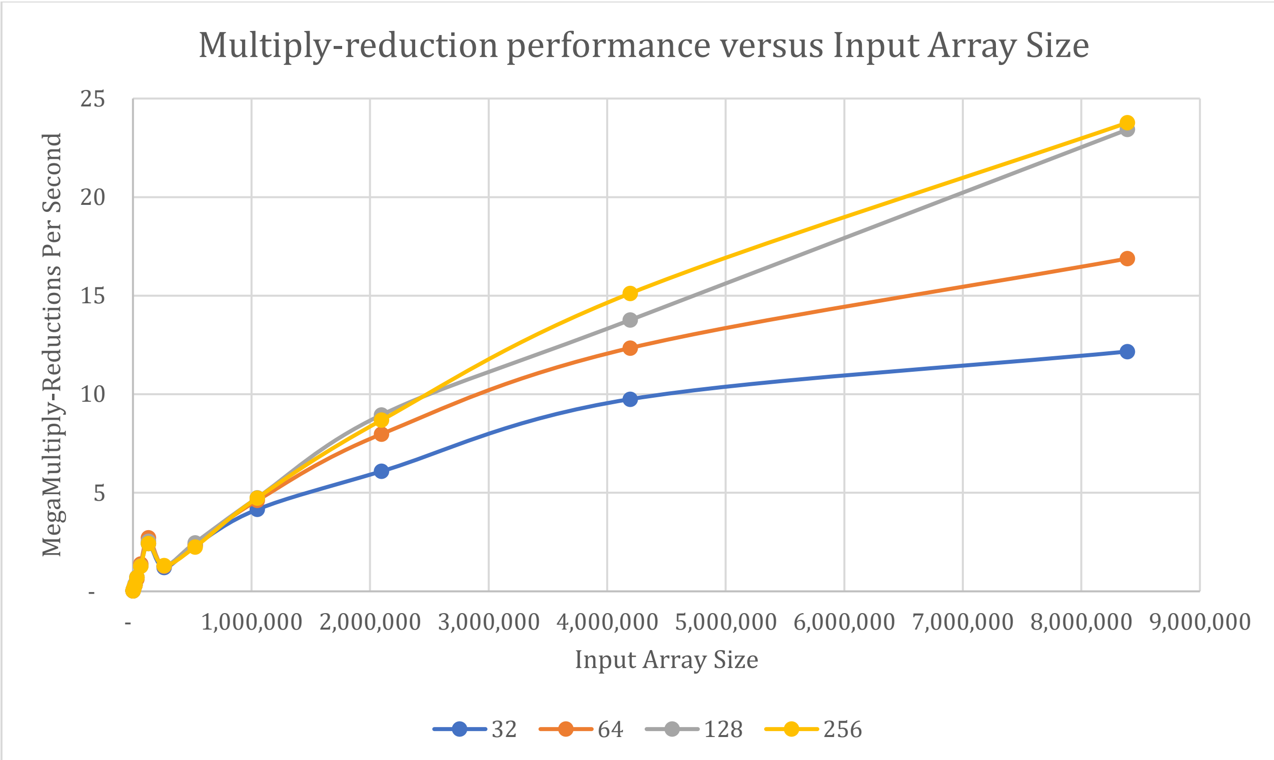
   Utilizing FMA instruction in OpenCL can boost performance of a multiplying and adding operation to the same speed as single operation such as multiplying.

   Using OpenCL with large Local Work Size and Global Dataset Size to achieve more benefits from GPU parallel computing.

3- Third Part

I ran this on DGX and below is the performance result:

	Local Work Size			
Input Array Size	32	64	128	256
1,024	0.021	0.020	0.020	0.023
2,048	0.045	0.045	0.043	0.041
4,096	0.087	0.090	0.090	0.091
8,192	0.179	0.180	0.163	0.182
16,384	0.321	0.317	0.362	0.320
32,768	0.706	0.628	0.727	0.705
65,536	1.308	1.392	1.284	1.267
131,072	2.424	2.714	2.537	2.426
262,144	1.204	1.273	1.295	1.305
524,288	2.334	2.390	2.465	2.252
1,048,576	4.155	4.611	4.747	4.736
2,097,152	6.090	7.980	8.954	8.684
4,194,304	9.749	12.345	13.765	15.118
8,388,608	12.165	16.879	23.426	23.772



4- Third Part Commentary:

a) What pattern are you seeing in this performance curve?

The larger the Local Work Size or the larger the Global Dataset Size, the higher the performance.

b) Why do you think the pattern looks this way?

For larger Local Work Size, each processing element can share memory and synchronize with other threads in the same work group so it will increase the performance.

For larger Global Dataset Size, it can utilize the GPU parallel computing benefits from OpenCL. If the Global Dataset Size is too small, the overhead cost of parallel computing will cancel the benefit.

c) What does that mean for the proper use of GPU parallel computing?

Using reduction to sum up the work group total inside the kernel will result in a much smaller array in the global memory comparing to the original global dataset size. Therefore, parallel computing of smaller sum inside each kernel can help to boost the calculation of total sum significantly.