

Holly Straley

straleyh@oregonstate.edu

CS475 – Spring 2018

Project 6

OpenCL Array Multiply, Multiply-Add, and Multiply-Reduce

1. Platform

I ran this program on the rabbit server.

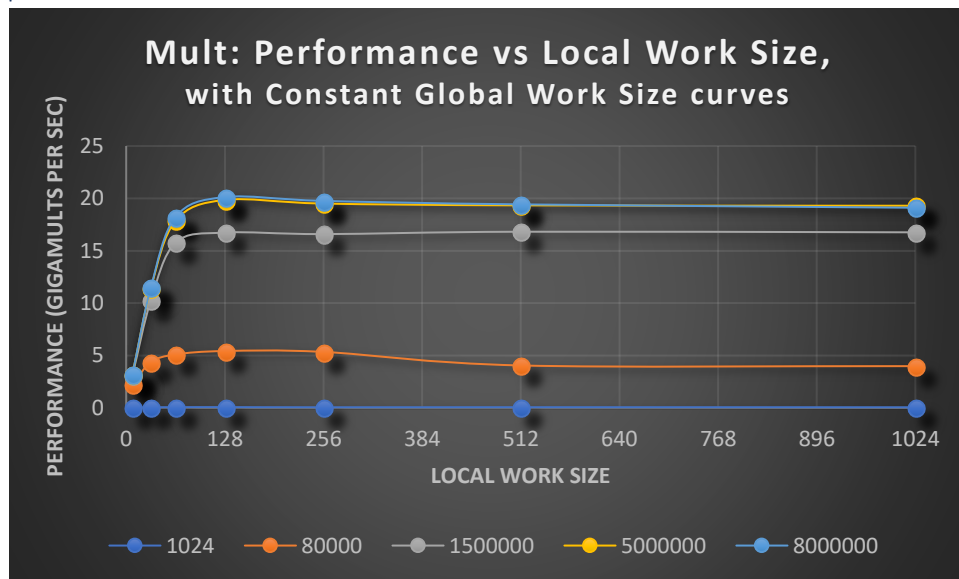
2. Performance Data

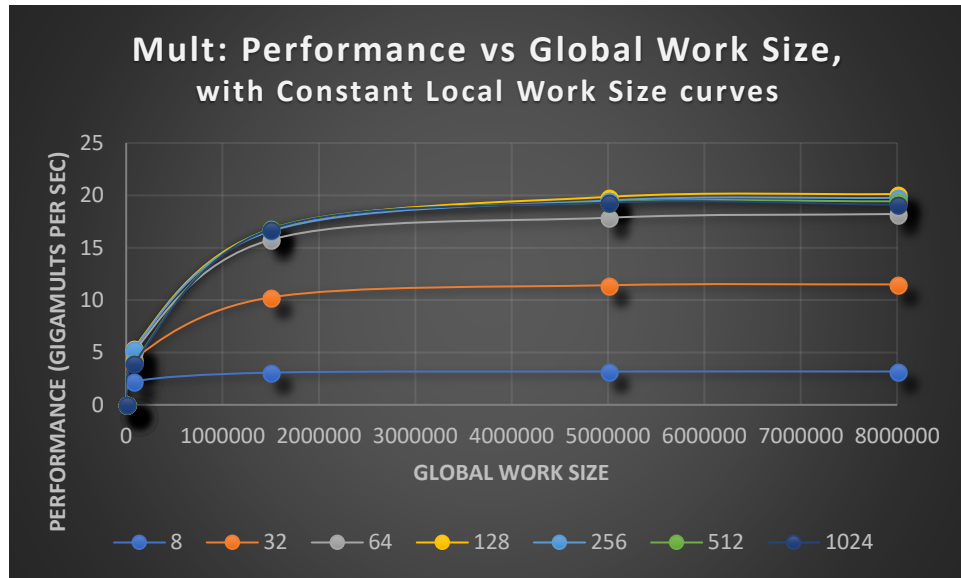
2.1 Part One: Multiply

2.1.1 Table

	8	32	64	128	256	512	1024
1024	0.051745	0.06153	0.071105	0.074347	0.059609	0.064264	0.060213
80000	2.213637	4.359541	5.098584	5.430188	5.344551	4.051276	3.986471
1500000	3.077921	10.26258	15.75954	16.73874	16.59238	16.81812	16.76108
5000000	3.180281	11.41898	17.86076	19.85204	19.50677	19.33471	19.30157
8000000	3.178921	11.49701	18.21818	20.12453	19.76172	19.43016	19.10891

2.1.2 Graphs



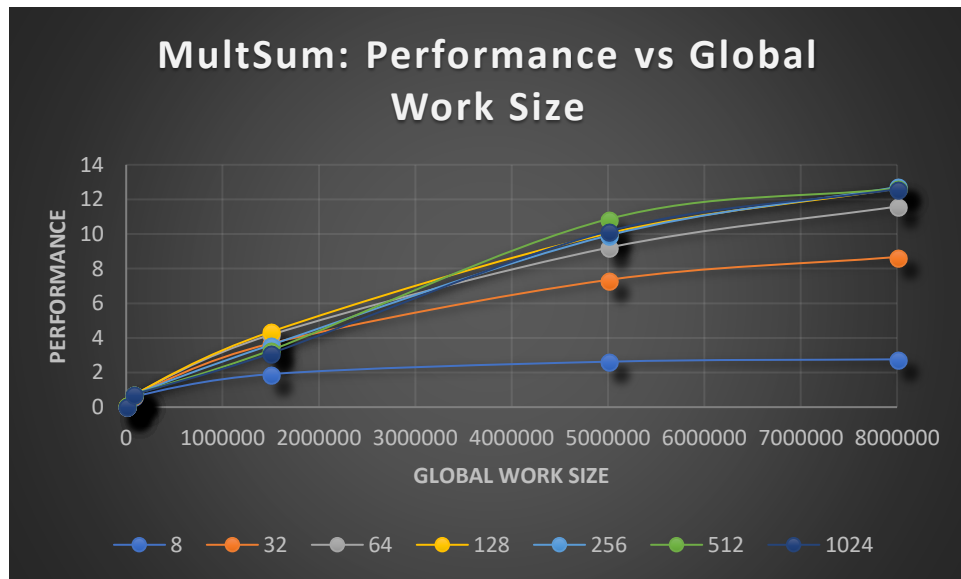
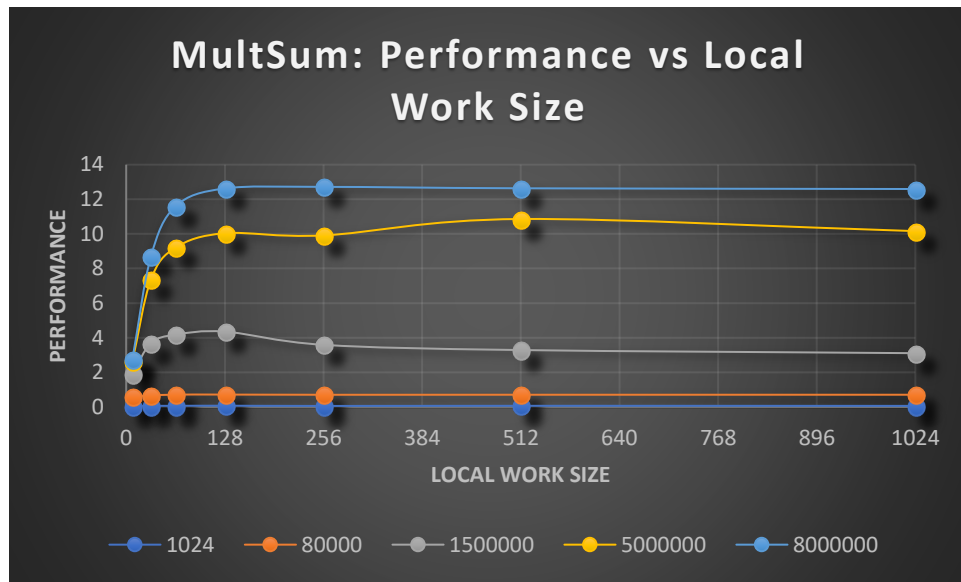


2.2 Part Two: Multiply and Sum

2.2.1 Table

	8	32	64	128	256	512	1024
1024	0.046721	0.067345	0.067721	0.076	0.060848	0.069701	0.064268
80000	0.604516	0.664855	0.714307	0.718507	0.706794	0.708284	0.708952
1500000	1.908102	3.678052	4.177181	4.354303	3.592699	3.290599	3.108745
5000000	2.626027	7.350663	9.206028	10.03608	9.912562	10.85952	10.15896
8000000	2.760407	8.675172	11.58753	12.62273	12.7166	12.63549	12.59314

2.2.2 Graphs



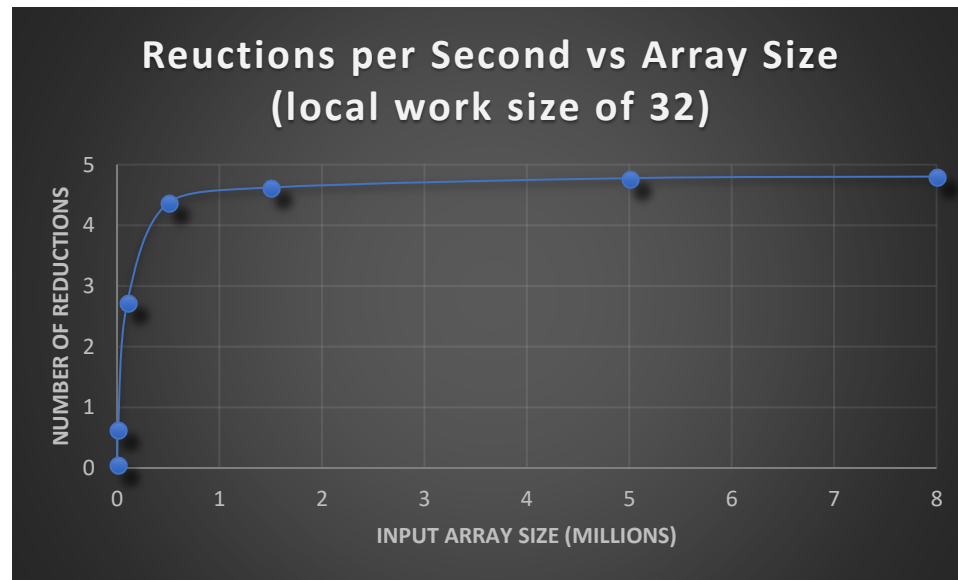
2.2 Part Three: Multiply and Reduce

2.2.1 Table

	32
1000	0.059884
10000	0.627143

100000	2.733431
500000	4.36823
1500000	4.623694
5000000	4.77723
8000000	4.803349

2.2.2 Graph



3 Patterns

3.1 Part One and Two

OpenCL performance increases greatly as local work group and global work groups sizes increase but performance levels off at a certain point. Optimal local work group size is 128 and global work group size was optimal at or above 6500000.

3.2 Part Three

Performance greatly increases as input array size increases, but performance starts to level off with an input array size above 2 million.

3.3 Pattern Analysis

The data followed expected patterns in that performance increased as workgroup or array size increased and then performance “tops out” at a certain point. The slight decrease in performance between local work group sizes of 128 and 256 is likely due to CPU instructions causing a larger impact in performance on the GPU.

4 GPU Parallel Computing Analysis

GPUs work best for parallel programming with very large data sets that are doing single operations and using regular data structures and regular flow control. The above graphs show that using GPU for

parallel computing using small work groups or small arrays does not give great performance but if we want to are using group size or array sizes that are among the “leveled off” portions of the graphs, we can expect consistently greatly increased performance.