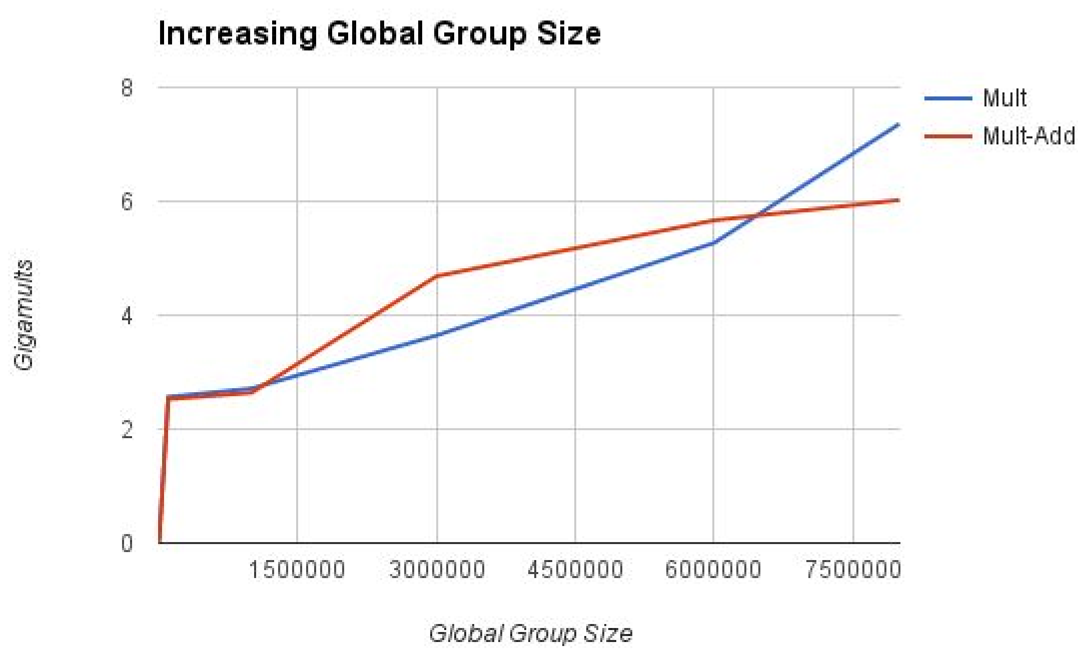
Cody Malick
CS 475
Assignment 6

Part 1

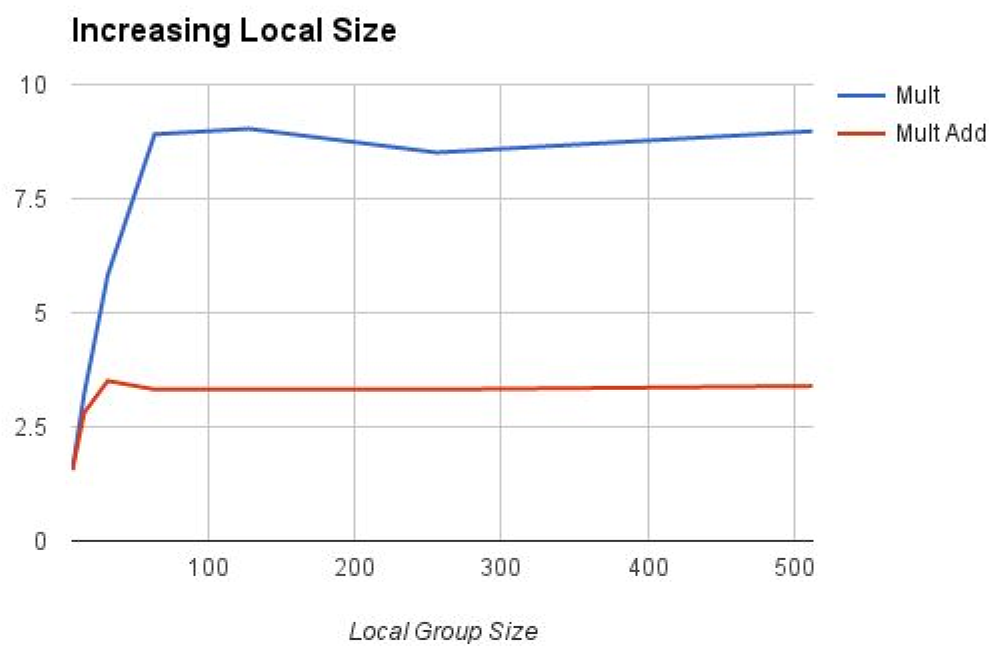
1. I ran this on my home desktop. It runs on an i5-2500K, eight gigabytes of ram, with a GTX 660ti, on Ubuntu 14.04.
2. Global Work Size

Global Group Size	Mult	Mult-Add
1000	0.024017	0.023794
10000	0.268053	0.234406
100000	2.564892	2.525189
1000000	2.712762	2.639421
3000000	3.646135	4.685765
6000000	5.271796	5.667098
8000000	7.359435	6.022568



Local Work Size

Local Size	Mult	Mult Add
8	1.574793	1.547659
16	3.225104	2.807741
32	5.820403	3.506605
64	8.910926	3.319123
128	9.031525	3.323276
256	8.510636	3.318581
512	8.976971	3.394033

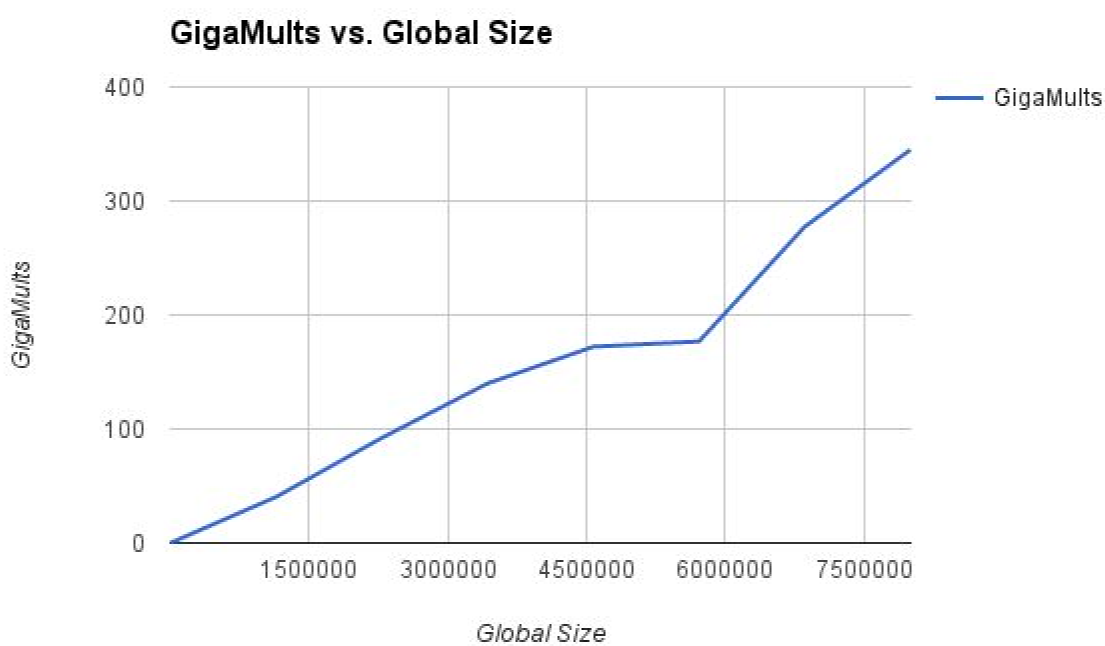


3. Overall, increasing the size of either group increases the performance. Increasing the local size increases the performance much faster than increasing the global size. Increasing the global size takes much longer to become effective.
4. As we increase the size of the work groups, we’ll see more performance out of each CUDA core because we’re fully utalizing the parallel ability of the video card.
5. The big difference between mult and mult-add is that mult-add has to do a manual, non-distributed addition of large amounts of numbers. Whereas with mult, we get full speedup without needing to add them up.
6. Proper use of GPU parallel computing would require not doing manual additions that stop the entire device from fully parallelizing a job. Every process running on it should be fully parallelized.

Part 2

1.

Global Size	GigaMults
1000	0.0308176
1143714	40.4525
2286428	92.243
3429142	139.999
4571856	172.159
5714570	176.801
6857284	277.454
7999998	344.932



2. As the global work group size increases, we see a somewhat linear increase in performance.
3. I think it is this way because as we increase the work group size, we get better utilization of the graphic cards parallel ability.
4. We want to use the largest possible global size to get the best possible performance.