Name: Venkata Vasantha Phani Sai

Email:paramkuv@oregonstate.edu

Project: Project06

Onid: paramkuv

Class: CS_575

1. What machine you run this on?

Ans. Rabbit.

2. Show the tables and graphs?

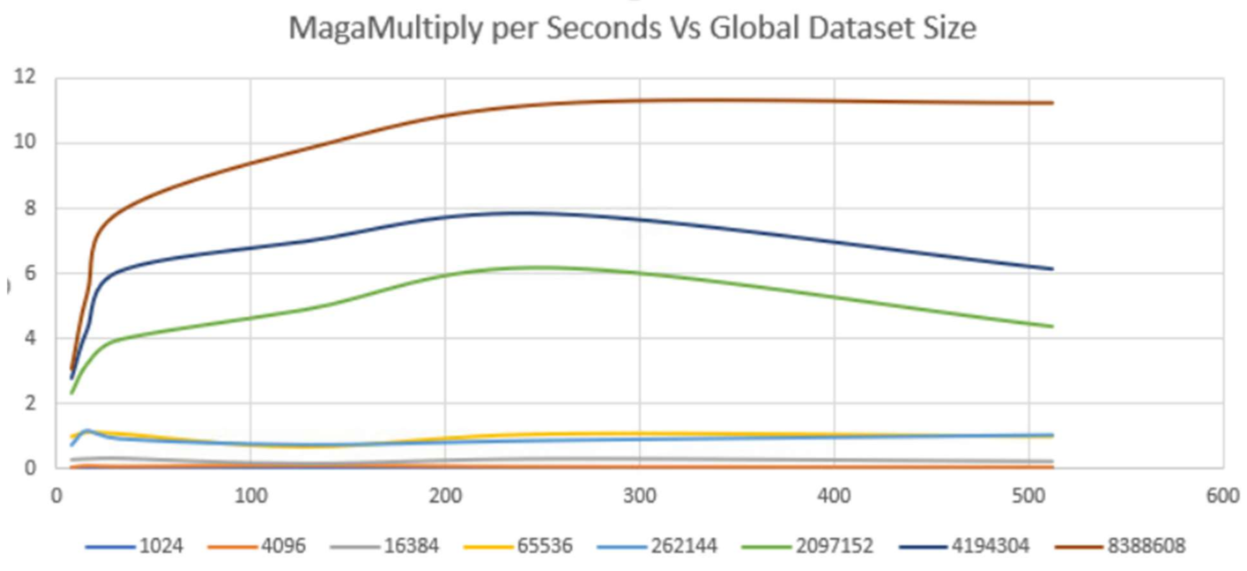
Ans.

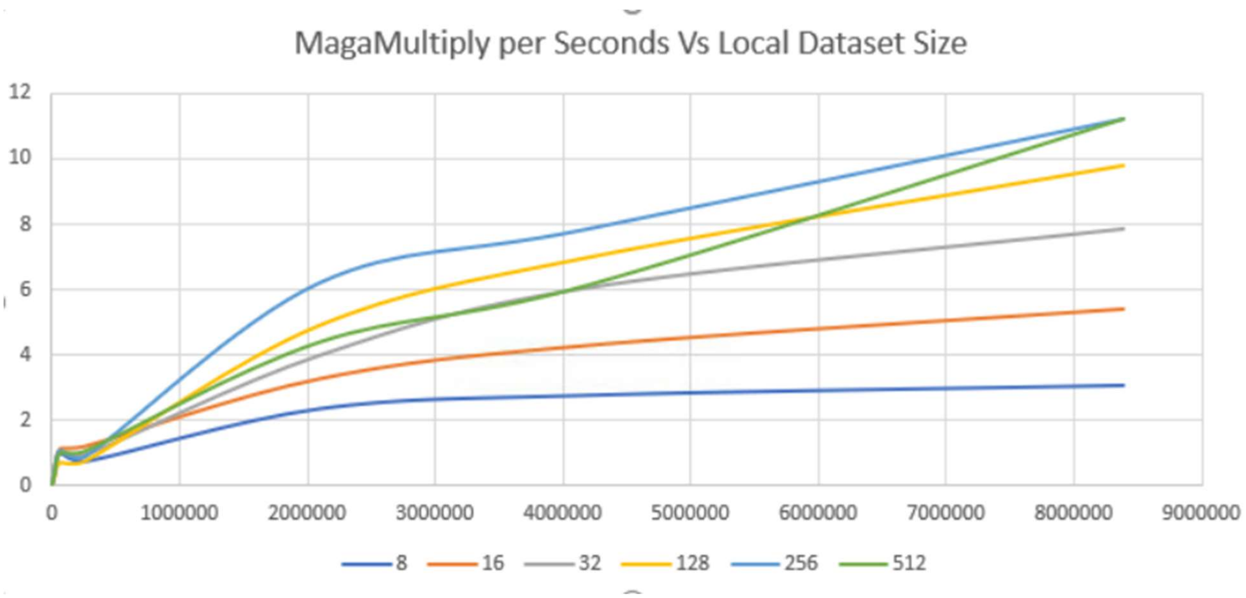
Multiplication Performance Results

Global_Size	LOCAL_SIZE	GigaMultsPerSecond
1024	8	0.017
1024	16	0.015
1024	32	0.012
1024	128	0.018
1024	256	0.017
1024	512	0.017
4096	8	0.043
4096	16	0.075
4096	32	0.062
4096	128	0.082
4096	256	0.057
4096	512	0.049
16384	8	0.269
16384	16	0.289
16384	32	0.303
16384	128	0.167
16384	256	0.291
16384	512	0.224
65536	8	0.973
65536	16	1.1

65536	32	1.058
65536	128	0.682
65536	256	1.053
65536	512	0.98
262144	8	0.72

	1024	4096	16384	65536	262144	1048576	2097152	4194304	8388608
8	0.017	0.043	0.269	0.973	0.72	1.984	2.34	2.754	3.058
16	0.015	0.075	0.289	1.1	1.176	2.694	3.24	4.259	5.374
32	0.012	0.062	0.303	1.058	0.922	3.917	3.975	6.051	7.863
128	0.018	0.082	0.167	0.682	0.731	0.149	4.907	6.992	9.824
256	0.017	0.057	0.291	1.053	0.867	0.275	6.198	7.846	11.224
512	0.017	0.049	0.224	0.98	1.038	2.78	4.389	6.137	11.251





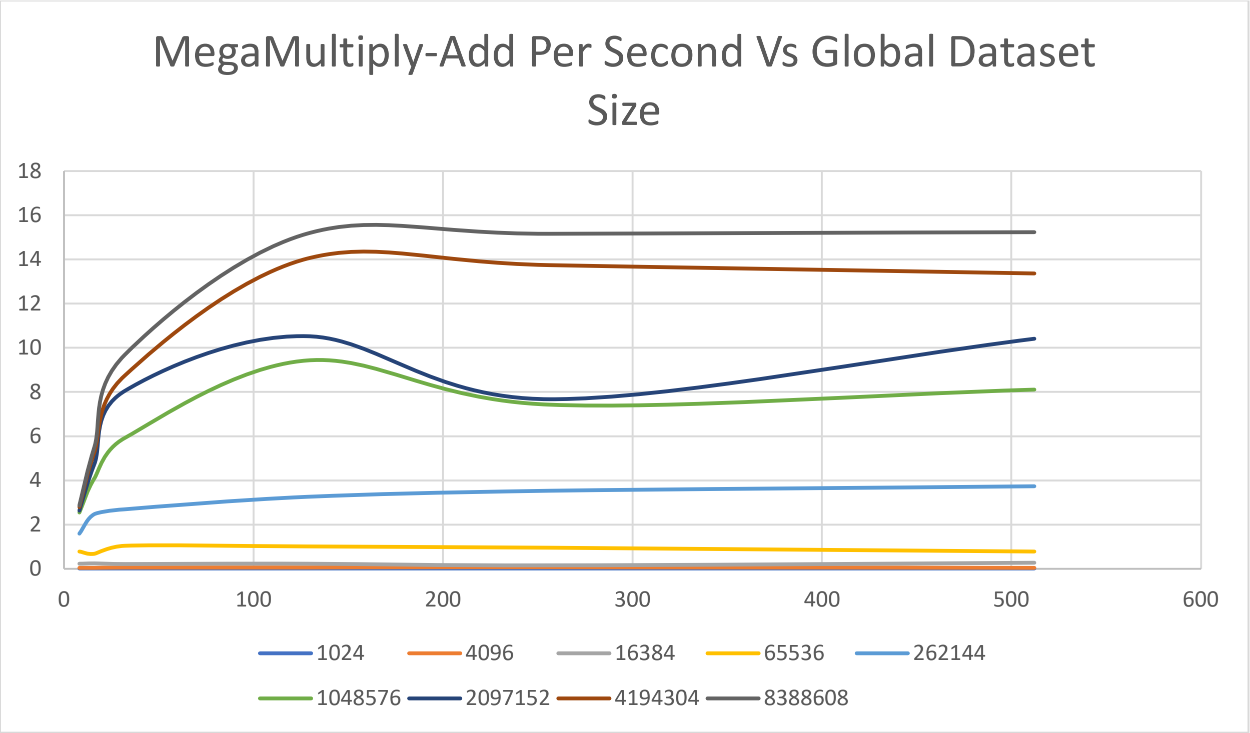
Multiplication-Add Performance Results

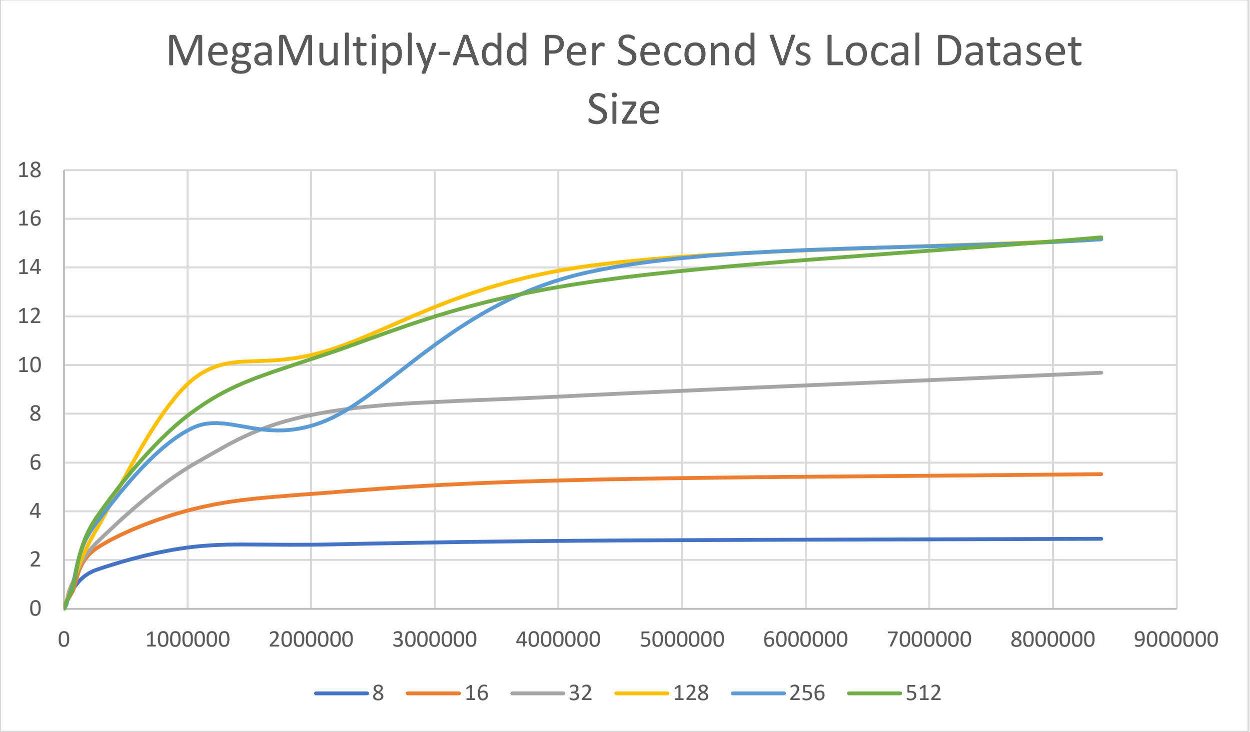
NMB	LOCAL_SIZE	GigaMultsPerSecond
1024	8	0.019
1024	16	0.012
1024	32	0.017
1024	128	0.014
1024	256	0.014
1024	512	0.017
4096	8	0.05
4096	16	0.045
4096	32	0.057
4096	128	0.069
4096	256	0.072
4096	512	0.055
16384	8	0.237
16384	16	0.252
16384	32	0.222
16384	128	0.235
16384	256	0.16
16384	512	0.282
65536	8	0.793
65536	16	0.69

65536	32	1.048
65536	128	1.019
65536	256	0.955
65536	512	0.791
262144	8	1.6
262144	16	2.479
262144	32	2.704
262144	128	3.26
262144	256	3.535
262144	512	3.737
1048576	8	2.547
1048576	16	4.096
1048576	32	5.928
1048576	128	9.437
1048576	256	7.438
1048576	512	8.12
2097152	8	2.644
2097152	16	4.758
2097152	32	8.054
2097152	128	10.54
2097152	256	7.684
2097152	512	10.41
4194304	8	2.796
4194304	16	5.294
4194304	32	8.752
4194304	128	14.031
4194304	256	13.755
4194304	512	13.371
8388608	8	2.869
8388608	16	5.523
8388608	32	9.686
8388608	128	15.165
8388608	256	15.163
8388608	512	15.246

	1024	**4096**	**16384**	**65536**	**262144**	**1048576**	**2097152**	**4194304**	**8388608**
8	0.019	0.05	0.237	0.793	1.6	2.547	2.644	2.796	2.869
16	0.012	0.045	0.252	0.69	2.479	4.096	4.758	5.294	5.523
32	0.017	0.05	0.22	1.04	2.704	5.928	8.054	8.752	9.686

		7	2	8					
128	0.014	0.069	0.235	1.019	3.26	9.437	10.54	14.031	15.165
256	0.014	0.072	0.16	0.955	3.535	7.438	7.684	13.755	15.163
512	0.017	0.055	0.282	0.791	3.737	8.12	10.41	13.371	15.246





3. What patterns are you seeing in the performance curves?

Ans. Both parts of the graphs have a raise in performance by an increase in the size of the global and local dataset size. In short, the higher the size of the global and local dataset size the performance would be better according to this graph. Except for the MegaMultiply-Add Per Second Vs Local Dataset Size where the performance of 128, 256, and 512 are joined into one, this is slightly unusual behavior, and this is caused due to the load on the system as well as in MegaMultiply Per Second Vs Local Dataset Size the 512 perform is grown in a similar way as other graph lines.

4. Why do you think the patterns look this way?

Ans.

Global Dataset Size:

Performance increases with the increase in the size of the Global dataset because if the Global dataset is larger than that means that the maximum utilization of the GPU Parallelism computing the benefits of the OpenCL. If the Global dataset size is very small, then the benefits of parallel computing cant are utilized due to the higher overhead costs.

Local Dataset size: Dataset Size: Performance increases with the increase in the size of the Local dataset because each processing element will share the memory and work with the threads in the group so the graph will show the increase in performance. In MegaMultiply-Add Per Second Vs Local Dataset Size where the performance of 128, 256, and 512 are joined into one, this is slightly unusual behavior, and this is caused due to the load on the system as well as in MegaMultiply Per Second Vs Local Dataset Size the 512 perform is grown in a similar way as other graph lines.

5. What is the performance difference between doing a Multiply and doing a Multiply-Add?

Ans.

When we compare both graphs of Multiply and Multiply-add both graphs show a similar kind of performance except for the graph in the multiplication is very slightly performing well. It means that both multiplication and multiplication-add use similar FMA instructions. But in the MegaMultiply-Add Per Second Vs Local Dataset Size the value of 512, 256, and 128 graphs are merged into the same point after 4000000. This behavior is very odd, and this is caused due to the load on the machine.

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

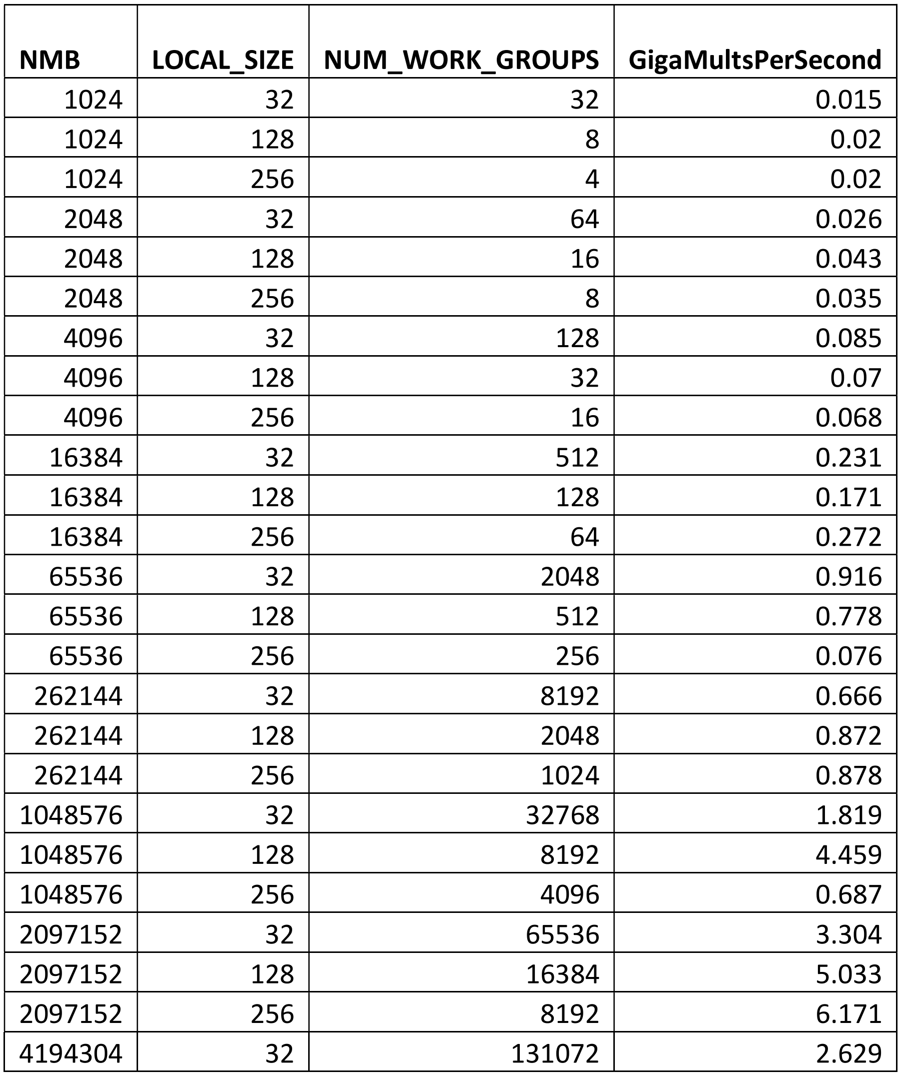
Ans.

The proper use of GPU parallel computing using the FMA instruction in the OpenCL to boost the performance of the multiplication and addition operations. This can be accomplished by using larger Global and Local Datasets sizes.

Part II

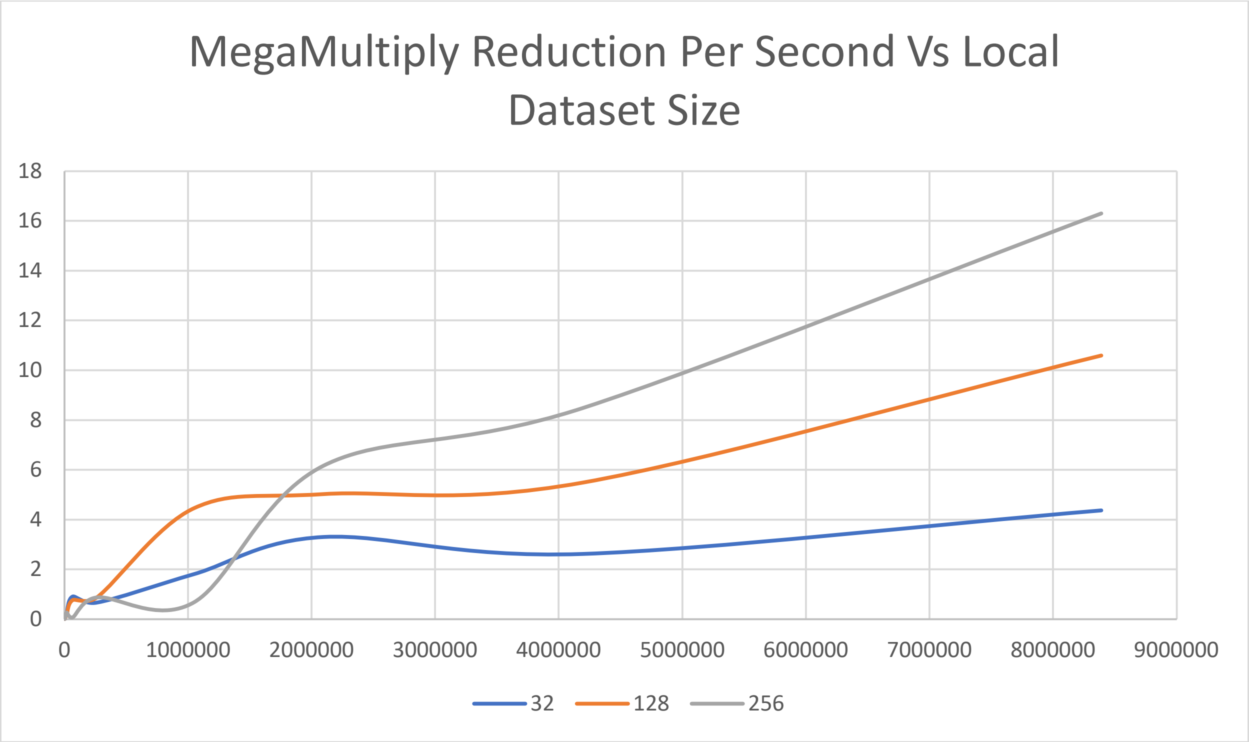
1. Show this table and graph?

Ans.



NMB	LOCAL_SIZE	NUM_WORK_GROUPS	GigaMultsPerSecond
1024	32	32	0.015
1024	128	8	0.02
1024	256	4	0.02
2048	32	64	0.026
2048	128	16	0.043
2048	256	8	0.035
4096	32	128	0.085
4096	128	32	0.07
4096	256	16	0.068
16384	32	512	0.231
16384	128	128	0.171
16384	256	64	0.272
65536	32	2048	0.916
65536	128	512	0.778
65536	256	256	0.076
262144	32	8192	0.666
262144	128	2048	0.872
262144	256	1024	0.878
1048576	32	32768	1.819
1048576	128	8192	4.459
1048576	256	4096	0.687
2097152	32	65536	3.304
2097152	128	16384	5.033
2097152	256	8192	6.171
4194304	32	131072	2.629

4194304	128	32768	5.491
4194304	256	16384	8.491
8388608	32	262144	4.371
8388608	128	65536	10.597
8388608	256	32768	16.309



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

Ans.

As the Global dataset size and Local dataset size increases the performance of the graphs are increasing.

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

Ans.

Similar to what I explained in the part-1 4th answer as the global dataset and local datasets increase the performance of the graphs will be increased due to the utilization of GPU parallel computing.

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

Ans.

By utilizing of the reduction to sum the workgroup instances will result in the smaller arrays in the global memory compared to the original global memory size. This results in better performance because the kernel can boost the calculation of the total sum significantly faster.