Susan Liu

933237062

CS 475

Professor Bailey

5/24/2021

Assignment 6

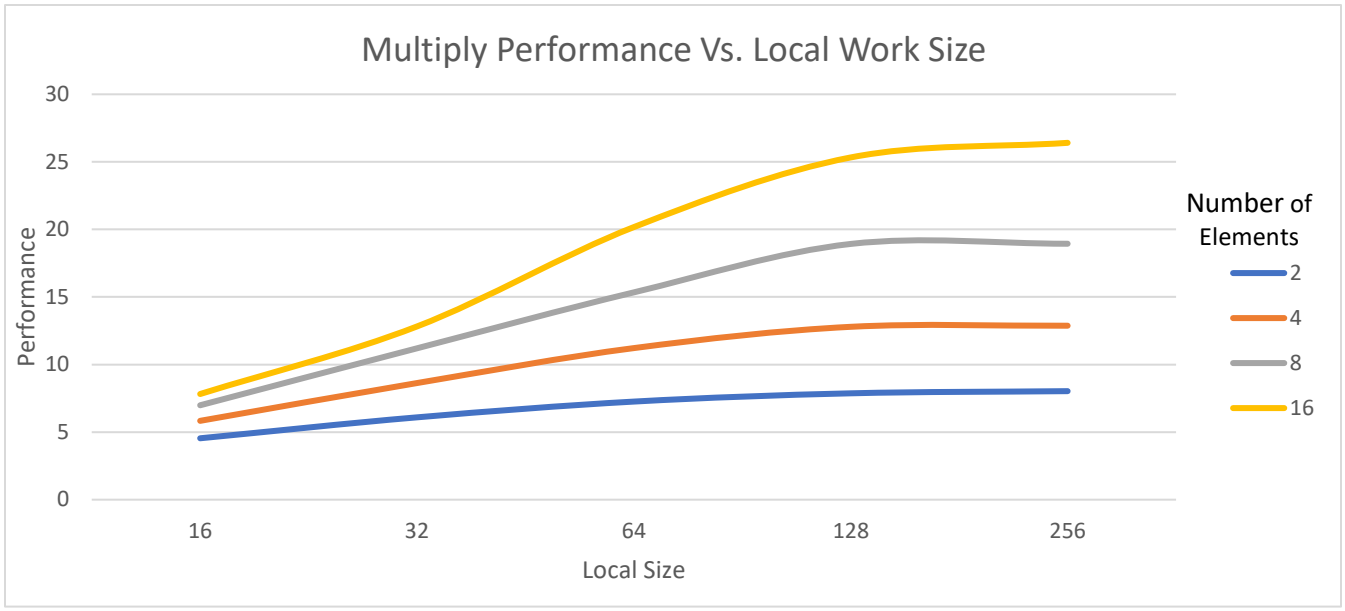
1. What machine you ran this on

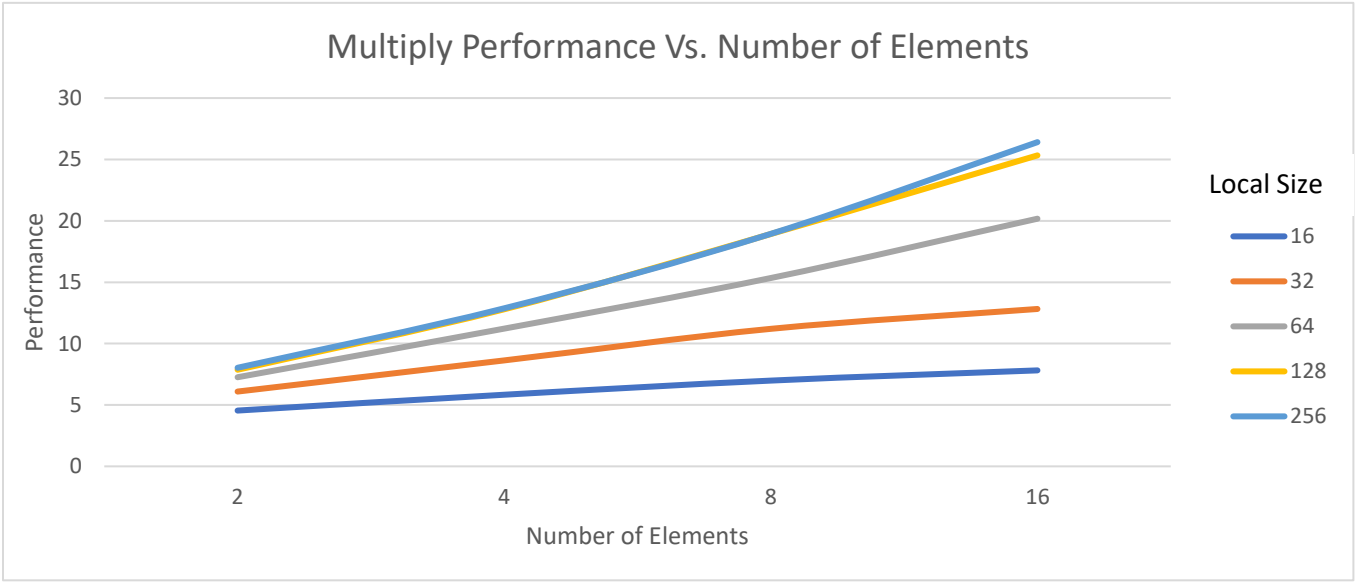
   DGX submit-b

2. Show the tables and graphs

Array Multiplication

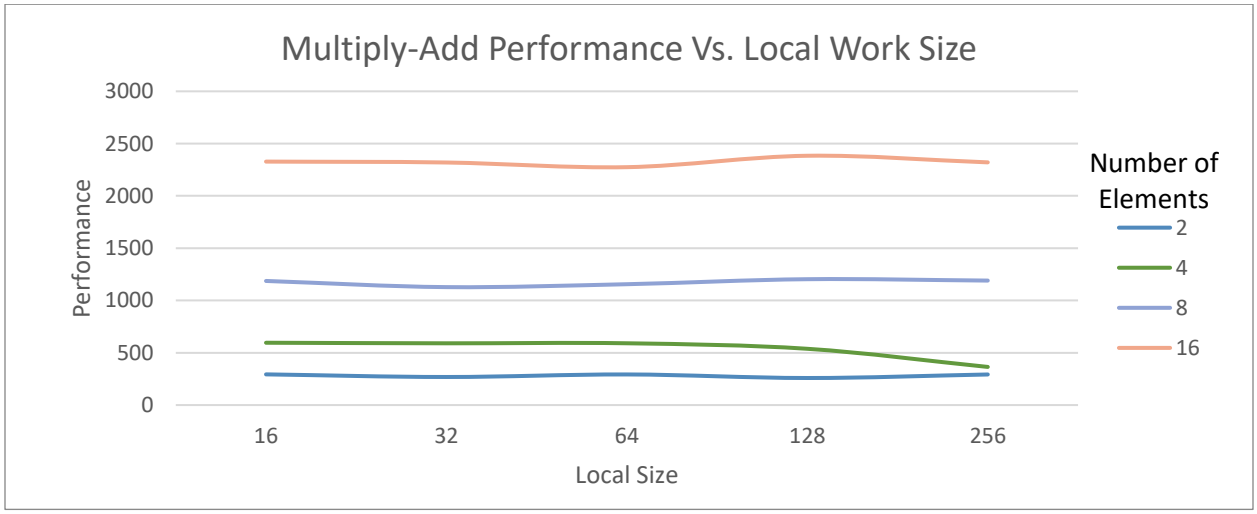
Number of Elements	Local Size	Speed (GigaMultsPerSecond)
2	16	4.544
2	32	6.093
2	64	7.26
2	128	7.87
2	256	8.031
4	16	5.834
4	32	8.631
4	64	11.219
4	128	12.794
4	256	12.875
8	16	6.987
8	32	11.203
8	64	15.342
8	128	18.927
8	256	18.94
16	16	7.819
16	32	12.82
16	64	20.174
16	128	25.329
16	256	26.409

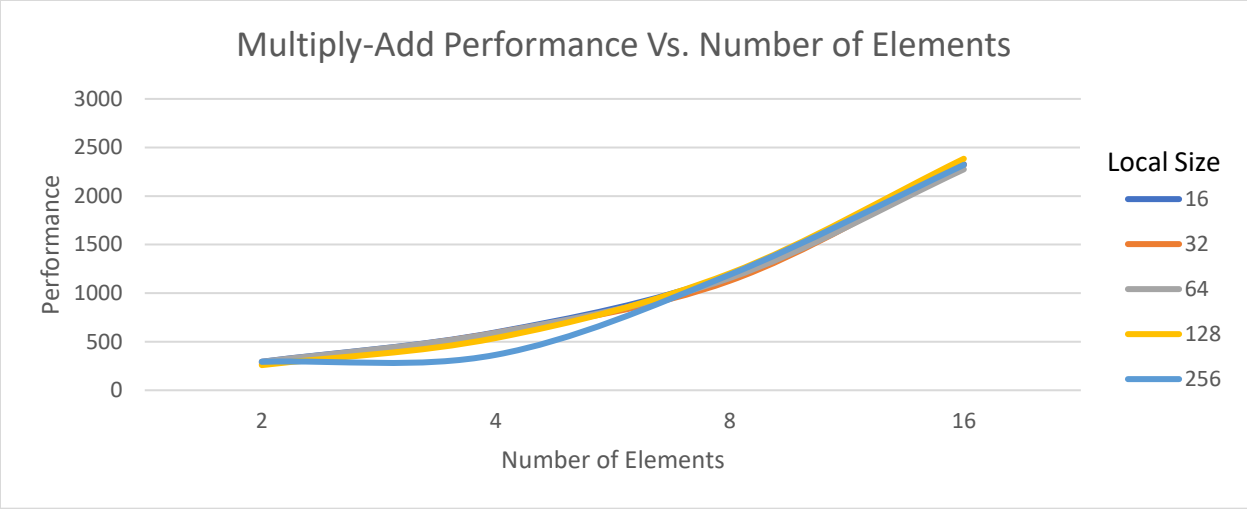

Multiply Performance Vs. Local Work Size
30
25
20
15
10
5
0
Performance
16
32
64
128
256
Local Size
Number of Elements
2
4
8
16


Multiply Performance Vs. Number of Elements
30
25
20
15
10
5
0
Performance
2
4
8
16
Number of Elements
Local Size
16
32
64
128
256

Array Multiplication Summation

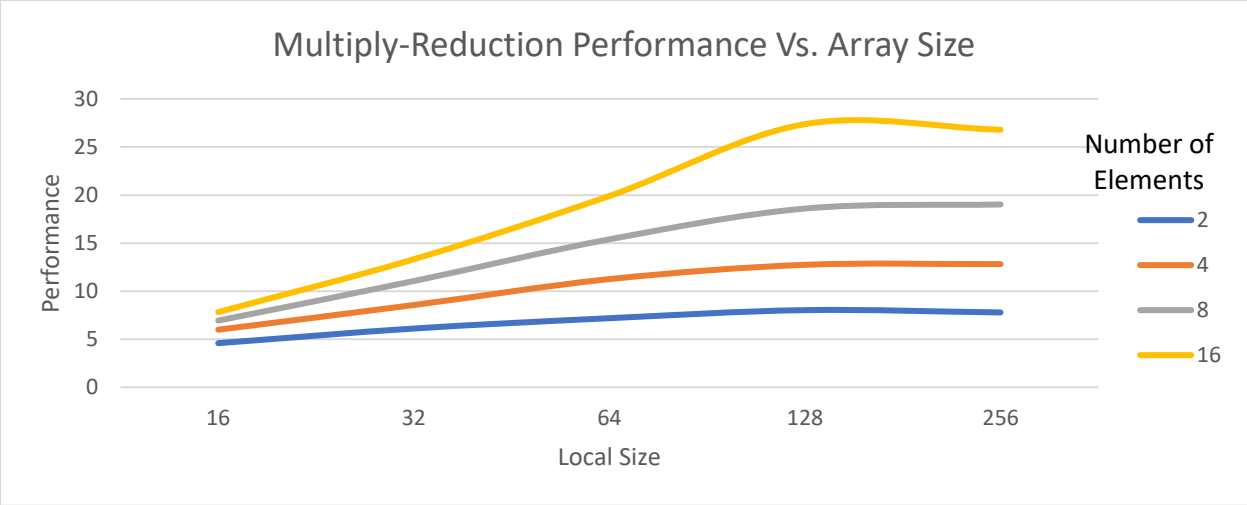
Number of Elements	Local Size	Speed (GigaMultsSumsPerSecond)
2	16	293.624
2	32	268.519
2	64	292.746
2	128	258.768
2	256	292.48
4	16	596.03
4	32	590.636
4	64	590.791
4	128	538.064
4	256	364.782
8	16	1186.719
8	32	1127.874
8	64	1155.658
8	128	1203.688
8	256	1190.326
16	16	2328.043
16	32	2319.351
16	64	2274.832
16	128	2383.803
16	256	2320.546

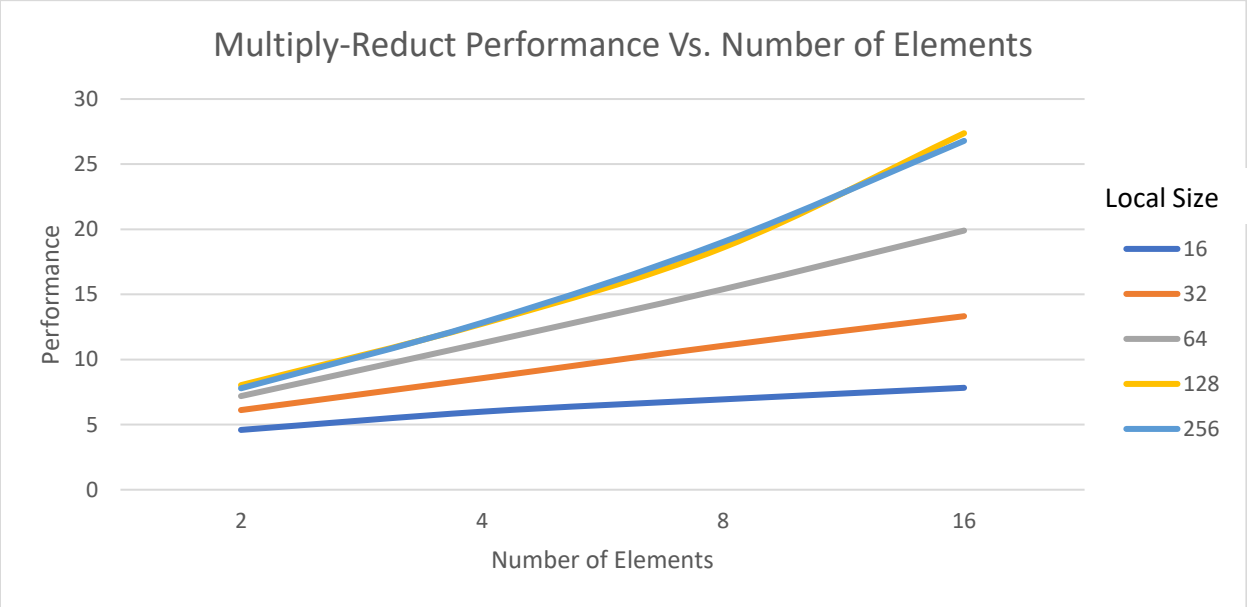




Array Multiplication Reduction

Number of Elements	Local Size	Speed (GigaMultsReductPerSecond)
2	16	4.593
2	32	6.115
2	64	7.188
2	128	8.03
2	256	7.789
4	16	5.99
4	32	8.563
4	64	11.254
4	128	12.741
4	256	12.816
8	16	6.939
8	32	11.055
8	64	15.394
8	128	18.593
8	256	19.021
16	16	7.826
16	32	13.324
16	64	19.898
16	128	27.375
16	256	26.787





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

   Looking at the Multiply performance vs. local work size graph, there is a constant increase in performance (Giga Mults per Second) to local size of 128 before the lines (number of elements) performance starts to flatten out. In this graph once the number of elements reach the local size of 256 the performance for bigger local sizes will be roughly similar to the performance of 256. Now

looking at the second Multiply graph, Multiply Performance vs. Number of Elements; as the number of elements increase the performance of the local size will increase linearly.

Looking at the Multiply-add performance vs. Local Work size, the performance (Giga Mults Sums per Second) is mostly constant as the local size increases as the number of elements increase, there might be a few shallow dips but in the next local size the dips return to the original performance. Looking at the second Multiply-add graph, Multiply-Add Performance vs. Number of Elements; as the number of elements increase there is a slow increase in performance.

Looking at the Multiply-Reduction Performance vs. Array Size, the performance (Giga Mults Reduct per Second) increases before the Number of element lines level out around the local size of 128. Looking at the second graph, Multiply-Reduction Performance vs. Number of Elements; there is a linear increase in the performance for the local size as the number of elements increases. Some of the increase is bigger than others such as the local size of 128 and 256 have a bigger decrease in performance as the performance time increases from 2 elements to 16 elements. While the local size of 256 has a much smaller decrease in performance as the performance time from 2 elements and 16 elements have a smaller difference.

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

   The reason the Multiply and Multiply-Reduction graphs increase before leveling out is because, when the lines levels out is the point where the efficiency has been maxed. So, the performance wont increase above or bellow that point.

   The reason the Multiply-Add performance is a relatively constant, is that it takes a lot of performance to run the code, so no matter the local size or the number of elements the performance for all local size will be similar.

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

   Well, the Multiply performance is in the 0-30 range while the Multiply-Add performance is in the 0-2500 range. The only difference is the curves, while the Multiply graph has a curve where the performance increases than levels out, while the Multiply-Add graph has a line that is constant, except for a few time the line makes a shallow dip.

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

   Being able to split up tasks to be able to parallelize into smaller more manageable sections. In the program we need to set both global and local size, so we have to take into consideration of register use and data size to maximize performance of the code, which could cause speed increase. Simply, the goal is to be able to efficiently do its work.