CS 475 Project 6

Aaron Frost
frostaa@oregonstate.edu
6/3/2023

1. Ran on rabbit
2. Here is the output in .csv form:

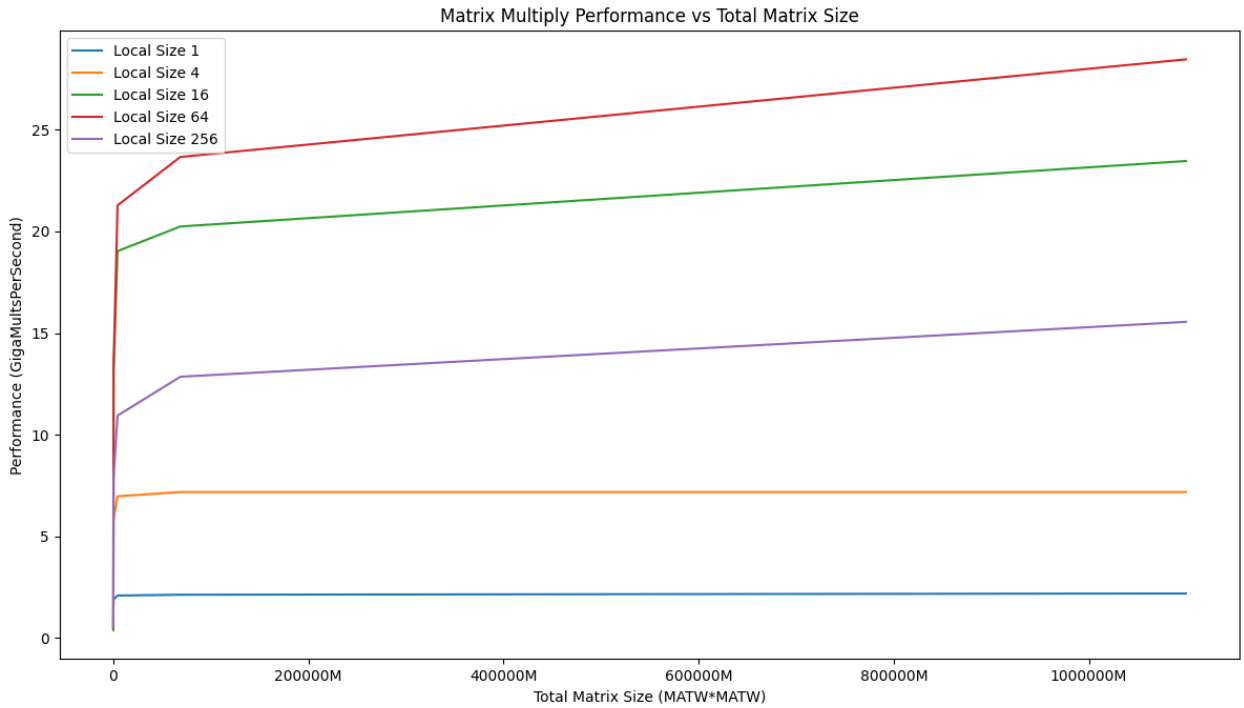
MATW	WORK_ELEMENTS	GigaMultsPerSecond
1024	1	0.44
1024	4	0.44
1024	16	0.39
1024	64	0.5
1024	256	0.54
4096	1	1.31
4096	4	2.52
4096	16	3.25
4096	64	4.01
4096	256	2.33
16384	1	1.9
16384	4	5.84
16384	16	13.1
16384	64	13.47
16384	256	7.93
65536	1	2.09
65536	4	6.97
65536	16	19.03
65536	64	21.28
65536	256	10.94
262144	1	2.13
262144	4	7.18
262144	16	20.24
262144	64	23.66
262144	256	12.85
1048576	1	2.19

1048576	4	7.18
1048576	16	23.46
1048576	64	28.46
1048576	256	15.55

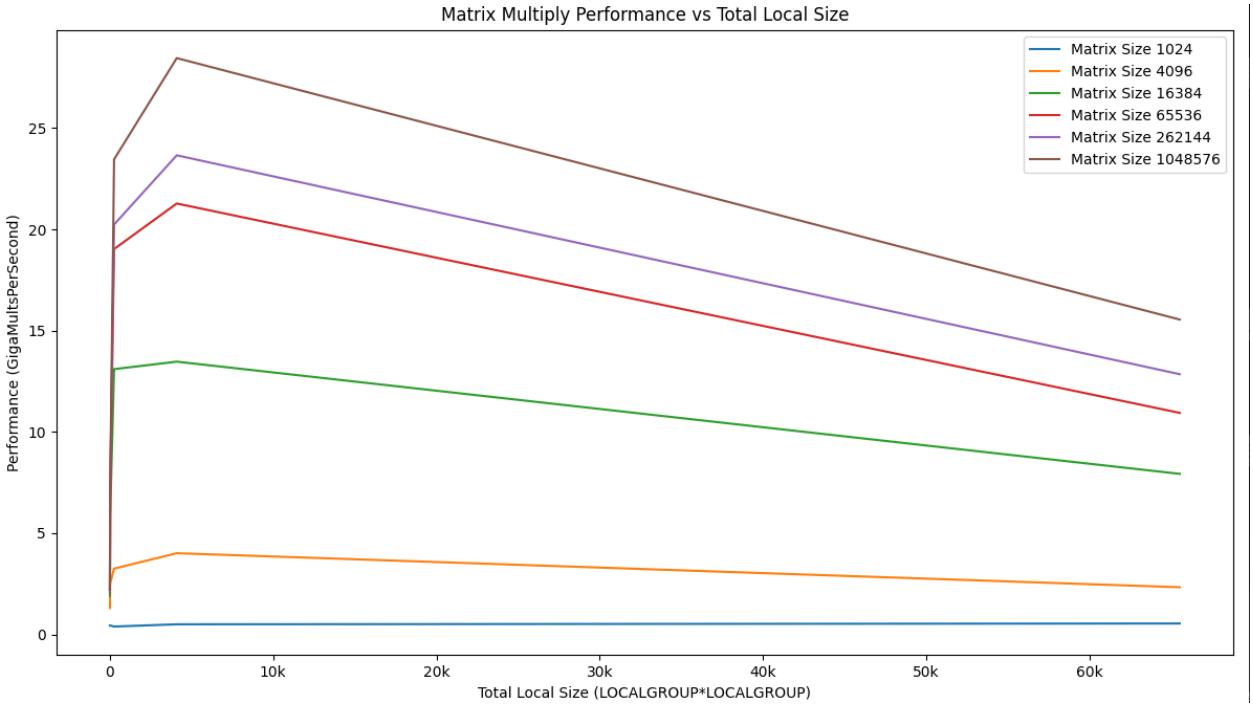
Here is the pivot table for performance (GigaMultsPerSecond):

	WORK_ELEMENTS				
MATW	1	4	16	64	256
1024	0.44	0.44	0.39	0.5	0.54
4096	1.31	2.52	3.25	4.01	2.33
16384	1.9	5.84	13.1	13.47	7.93
65536	2.09	6.97	19.03	21.28	10.94
262144	2.13	7.18	20.24	23.66	12.85
1048576	2.19	7.18	23.46	28.46	15.55

Here is the “Matrix multiply performance versus total matrix size (MATW*MATW), with a series of colored Constant-Local-Size curves”



Here is the “Matrix multiply performance versus total Local Size (LOCALGROUP*LOCALGROUP), with a series of colored Constant-Matrix-Size curves”



3.

The Matrix Multiply Performance vs Total Matrix Size:

Each performance curve Increases linearly with the total matrix size, and the curves with higher local size have higher performance overall, except for 256, which has about half the performance from 64.

In the Matrix Multiply Performance vs Total Local Size graph:

Each performance curve decreases linearly with the total work group size, and curves with higher matrix size have higher performance overall consistently than those with lower matrix size.

4. There is a linear increase of performance in the first graph because larger matrices take advantage of data parallelism, so most of the GPU processing units are utilized at the same time, where more multiplications can perform every second. I think that using a local size of 256 led to memory contention or inefficient use of memory due to the GPU’s smaller amount of local memory.

I think that the decrease in performance with larger local work groups in the second graph was also due to memory contention and an increased overhead from managing a large number of threads.