CS 575 -- Spring Quarter 2023
Project #6
OpenCL Matrix Multiplication

Kuan Ting Liu

liukuan@oregonstate.edu

June 4 2023

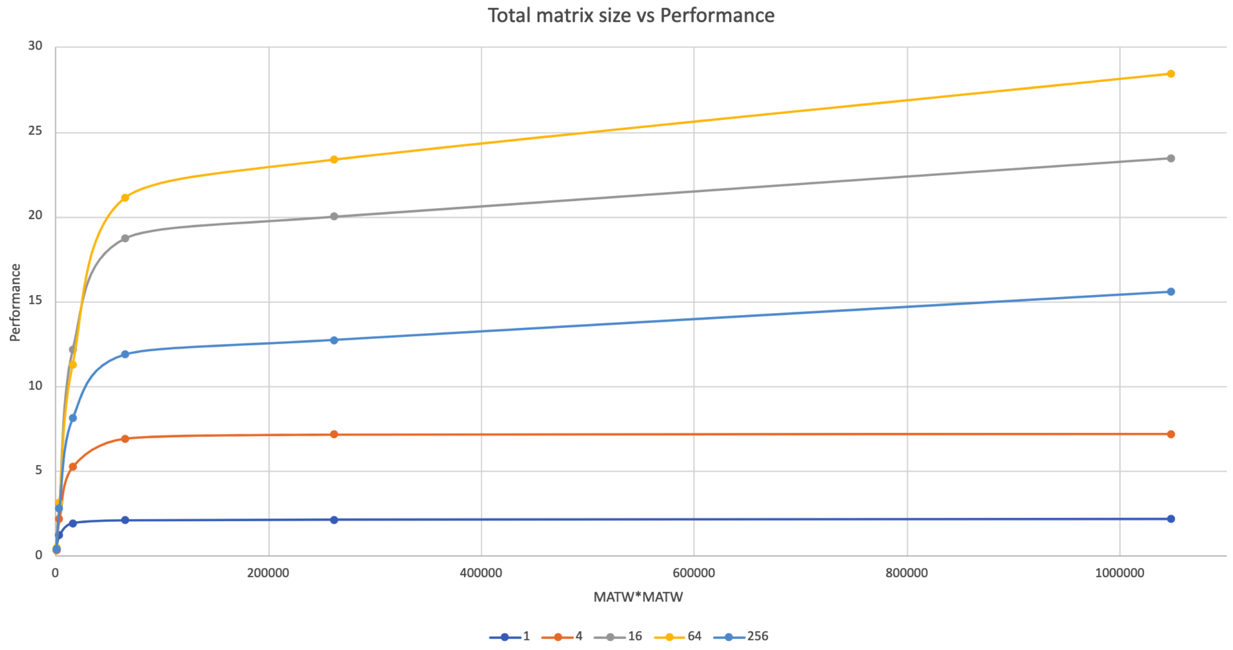
1. The running machine is rabbit.
2.

   The table:

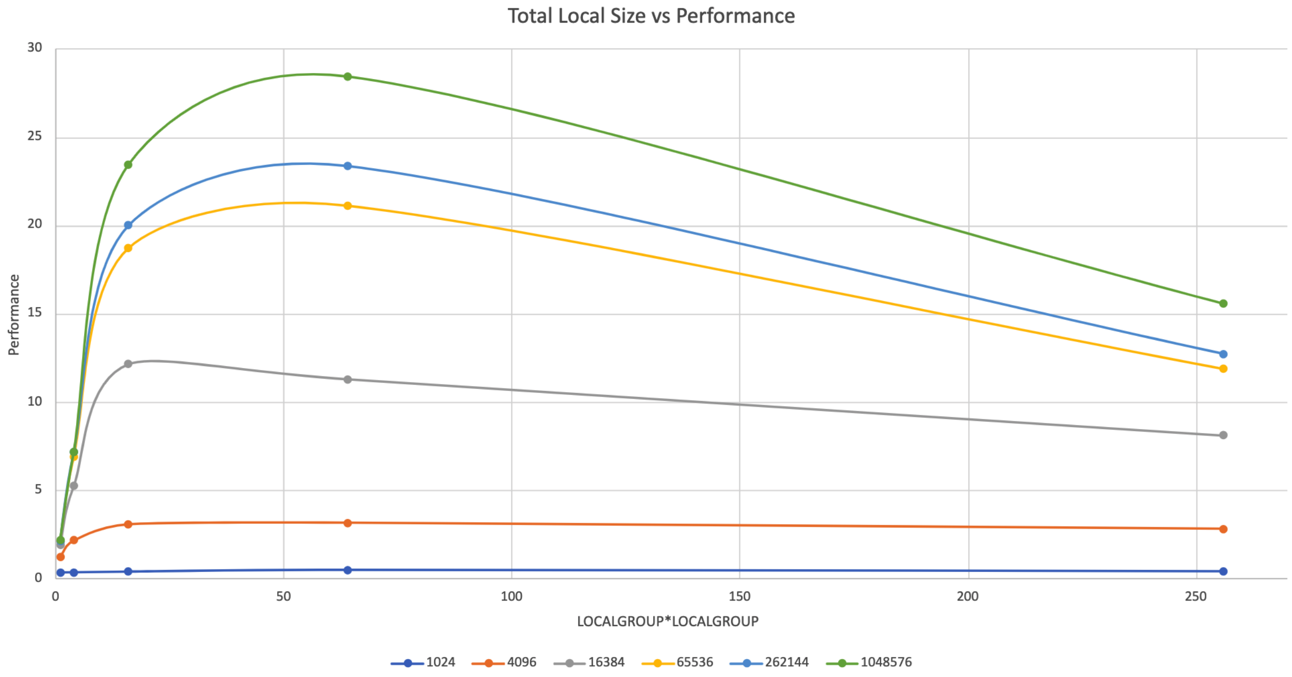
MATW*MATW	LOCALSIZE*LOCALSIZE	Performance
1024	1	0.34
1024	4	0.35
1024	16	0.39
1024	64	0.49
1024	256	0.41
4096	1	1.19
4096	4	2.14
4096	16	3.07
4096	64	3.16
4096	256	2.82
16384	1	1.92
16384	4	5.22
16384	16	12.13
16384	64	11.28
16384	256	8.09
65536	1	2.09
65536	4	6.9
65536	16	18.71
65536	64	21.11
65536	256	11.87
262144	1	2.13
262144	4	7.14
262144	16	19.99
262144	64	23.36
262144	256	12.73
1048576	1	2.17
1048576	4	7.18
1048576	16	23.44
1048576	64	28.43
1048576	256	15.57

The graphs:

Matrix multiply performance versus total matrix size (MATW*MATW), with a series of colored Constant-Local-Size curves:



Matrix multiply performance versus Total Local Size (LOCALGROUP*LOCALGROUP), with a series of colored Constant-Matrix-Size curves



3. In Matrix multiply performance versus Total matrix size, we can see that when the matrix size is increased, the performance will grow up. And when the matrix size is large enough (over 20000), the performance goes stable. The performance increases when the local size increases.
   In Matrix multiply performance versus total Local Size, we can see that when the local size increases, the performance increases. However, after the peak of local size of 50, the performance starts to decrease at size of 64. The performance increases when the total matrix size increases.

4. I think when using the OpenCL to do the calculation, the GPU memory is limited, and the globalworksize and localworksize all have some limitations. The threads and blocks can be assigned under a specific number for best performance. This will make the performance can not be increased when we keep increasing the local group size.