OpenCL Array Multiply, Multiply-Add, and Multiply-Reduce

Rui Gao

gaorui@oregonstate.edu

Project 6

CS 575

The cpp. Files are the same name, but they are ordered by the 1st 2nd 3rd situations.

1

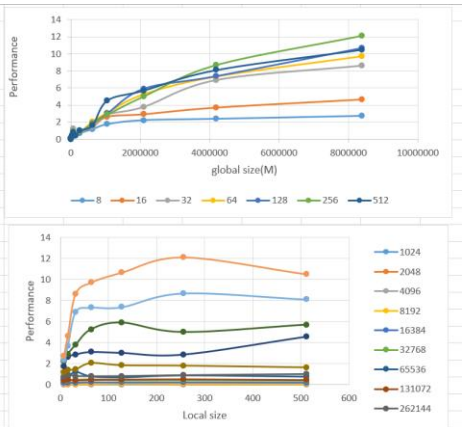
Rabbit. The files worked not good at DGX somehow, so I complete them at the Rabbit instead.

2

a.

1

global \ local	8	16	32	64	128	256	512
1024	0.011	0.013	0.013	0.014	0.013	0.013	0.013
2048	0.039	0.026	0.026	0.026	0.024	0.023	0.024
4096	0.05	0.059	0.055	0.054	0.045	0.056	0.048
8192	0.114	0.099	0.1	0.11	0.134	0.115	0.108
16384	0.21	0.208	0.181	0.234	0.213	0.225	0.202
32768	0.424	0.432	0.461	0.504	0.432	0.44	0.41
65536	0.653	0.746	1.236	0.817	0.72	0.899	0.792
131072	0.426	0.467	0.407	0.452	0.479	0.509	0.439
262144	0.719	0.938	0.821	0.796	0.812	0.901	1.016
624288	1.173	1.413	1.45	2.051	1.85	1.807	1.642
1048576	1.758	2.62	2.874	3.105	3.023	2.867	4.546
2097152	2.219	2.915	3.779	5.262	5.906	5.004	5.707
4194304	2.376	3.688	6.923	7.338	7.381	8.691	8.116
8388608	2.714	4.637	8.604	9.707	10.655	12.117	10.502

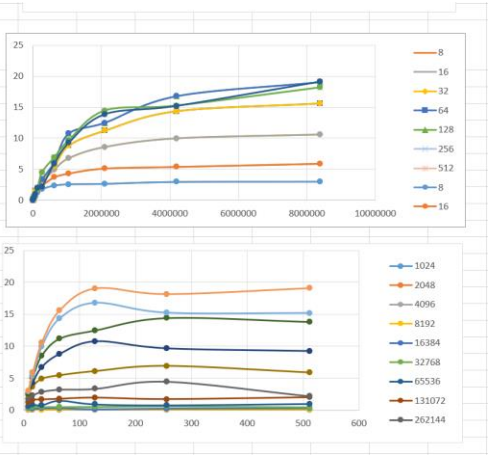


b

2

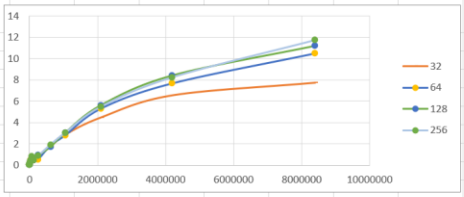
global \ local	8	16	32	64	128	256	512
1024	0.012	0.012	0.014	0.015	0.014	0.015	0.014
2048	0.032	0.04	0.024	0.028	0.029	0.032	0.028
4096	0.063	0.057	0.066	0.047	0.046	0.061	0.053
8192	0.104	0.112	0.089	0.114	0.127	0.104	0.111
16384	0.353	0.237	0.284	0.325	0.167	0.253	0.319
32768	0.374	0.525	0.475	0.467	0.494	0.587	0.42
65536	0.613	0.871	0.76	1.466	0.907	0.764	0.98
131072	1.215	1.604	1.66	1.763	1.936	1.715	1.989
262144	1.786	2.325	2.872	3.224	3.344	4.464	2.19
624288	2.354	3.693	4.889	5.444	6.1	6.919	5.885
1048576	2.561	4.308	6.75	8.792	10.778	9.668	9.249
2097152	2.638	5.116	8.543	11.203	12.488	14.439	13.837
4194304	2.941	5.389	9.939	14.35	16.786	15.241	15.185
8388608	2.987	5.9	10.561	15.608	19.061	18.164	19.09

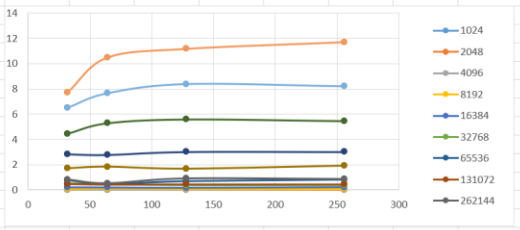
3



c.

3 local / global	32	64	128	256
1024	0.013	0.012	0.014	0.013
2048	0.025	0.022	0.028	0.025
4096	0.059	0.05	0.055	0.065
8192	0.1	0.133	0.105	0.101
16384	0.216	0.204	0.179	0.225
32768	0.538	0.477	0.39	0.394
65536	0.766	0.48	0.691	0.817
131072	0.495	0.45	0.462	0.461
262144	0.826	0.528	0.912	0.868
624288	1.735	1.844	1.706	1.914
1048576	2.823	2.768	3.021	3.023
2097152	4.445	5.29	5.604	5.471
4194304	6.528	7.668	8.406	8.231
8388608	7.713	10.49	11.195	11.73





3.

As the number of global size increase, the performance increases.

As the number of local size increase, the performance also increases.

4.

Because the" global size" is provided for all work groups and make it more paralleled (dividing works), so it is helpful.

As for the "local size", it is also more paralleled but inside.

5.

The multiply performance has a small hump when the global size up to 64K and the performance decreases later. But the multiply-add does not own a similar performance.

6.

I wanted to write that we'd better only use GPUs for simple and monotonous computations. But the Multiply-Add performance is higher.

The one with a reduction is similar with the first one. Maybe I misunderstand the meaning of requirements or the rabbit is an old rabbit now (DGX should be better).