CS 575

Project #6

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

Zhouxiang Meng、

mengz@oregonstate.edu

1. What machine you ran this on

   I run this project on OSU server.

2. Show the table and graph

   Multiply

	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288	1048576	2097152	4194304	8388608
8	0. 007	0. 015	0. 028	0. 041	0. 13	0. 244	0. 362	0. 585	1. 502	2. 204	3. 122	13. 769	25. 734	56. 134
16	0. 008	0. 012	0. 019	0. 056	0. 111	0. 214	0. 574	0. 976	1. 165	2. 06	2. 711	12. 551	25. 226	54. 487
32	0. 007	0. 01	0. 032	0. 049	0. 104	0. 315	0. 323	0. 825	1. 535	1. 959	3. 155	12. 019	26. 885	51. 106
64	0. 007	0. 014	0. 03	0. 071	0. 089	0. 242	0. 474	0. 719	1. 112	2. 168	3. 037	10. 915	28. 617	57. 395
128	0. 007	0. 011	0. 029	0. 059	0. 128	0. 238	0. 453	0. 982	1. 014	2. 225	3. 528	13. 955	26. 331	63. 456
256	0. 005	0. 01	0. 029	0. 048	0. 103	0. 219	0. 468	0. 734	1. 322	3. 734	4. 497	9. 154	22. 703	54. 927
512	0. 005	0. 015	0. 027	0. 049	0. 1	0. 265	0. 257	0. 76	1. 2	3. 628	5. 613	13. 806	31. 345	50. 702

   Multiply + Add

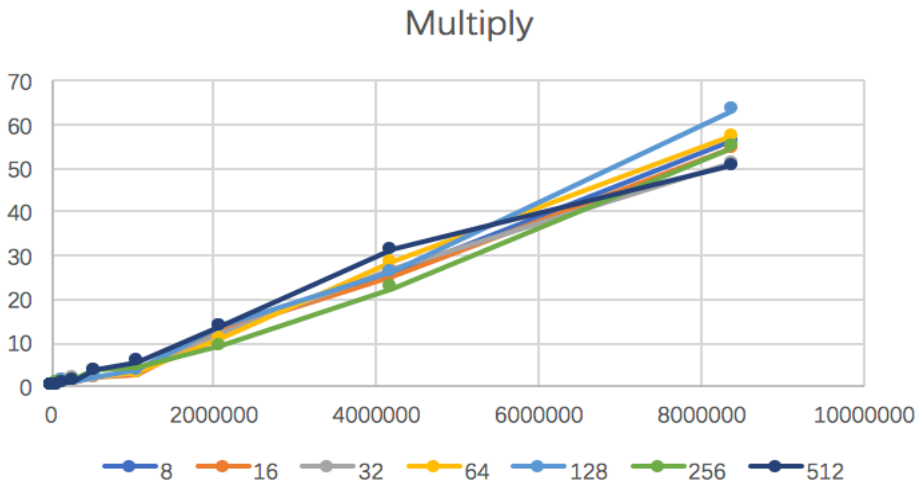
	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288	1048576	2097152	4194304	8388608
8	0. 006	0. 01	0. 029	0. 064	0. 109	0. 18	0. 393	0. 936	1. 735	3. 659	7. 017	11. 061	25. 477	59. 225
16	0. 007	0. 014	0. 023	0. 05	0. 136	0. 232	0. 401	0. 863	1. 703	3. 177	7. 579	14. 893	27. 537	64. 661
32	0. 007	0. 012	0. 037	0. 076	0. 128	0. 246	0. 411	1. 02	1. 774	3. 929	6. 704	14. 304	35. 597	64. 254
64	0. 006	0. 015	0. 026	0. 068	0. 127	0. 22	0. 428	0. 936	1. 716	3. 334	7. 468	15. 294	31. 243	56. 601
128	0. 008	0. 015	0. 038	0. 06	0. 121	0. 243	0. 46	0. 87	2. 047	3. 223	7. 095	15. 433	26. 399	60. 993
256	0. 008	0. 017	0. 027	0. 057	0. 133	0. 229	0. 459	1. 004	1. 759	3. 649	7. 557	13. 199	29. 613	59. 745
512	0. 008	0. 016	0. 031	0. 057	0. 123	0. 22	0. 393	0. 925	1. 878	3. 893	8. 429	11. 531	28. 885	62

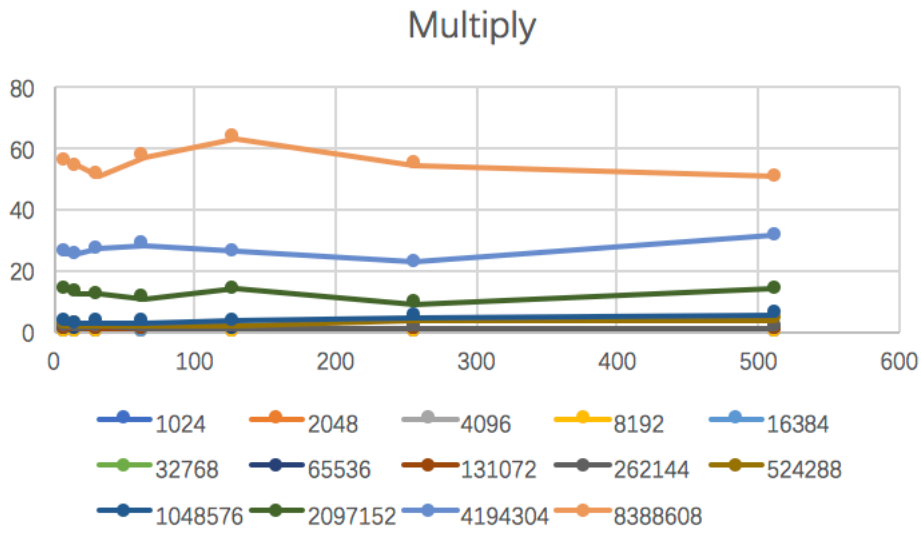
   Reduction

	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288	1048576	2097152	4194304	8388608
32	0. 006	0. 017	0. 033	0. 058	0. 1	0. 21	0. 393	0. 817	1. 895	4. 373	7. 235	18. 244	29. 96	55. 827

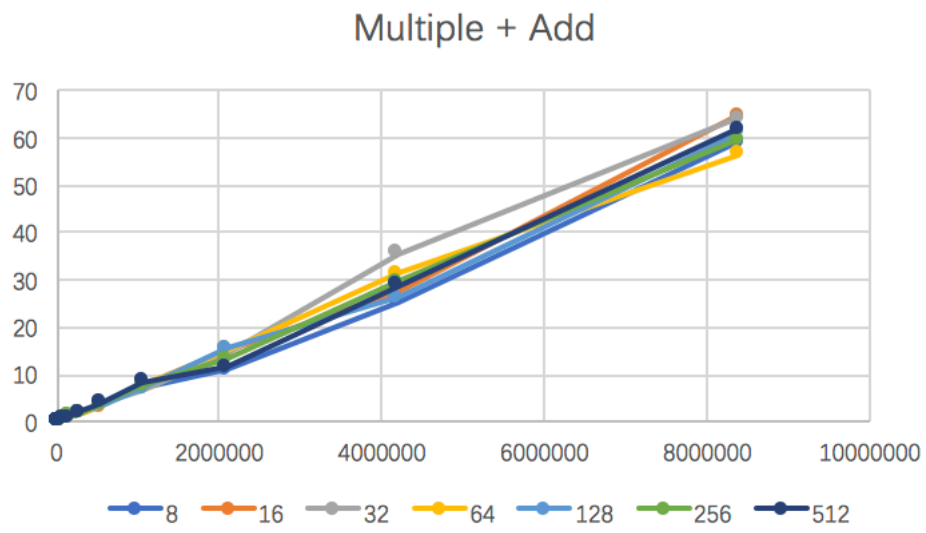
3. Graph of results

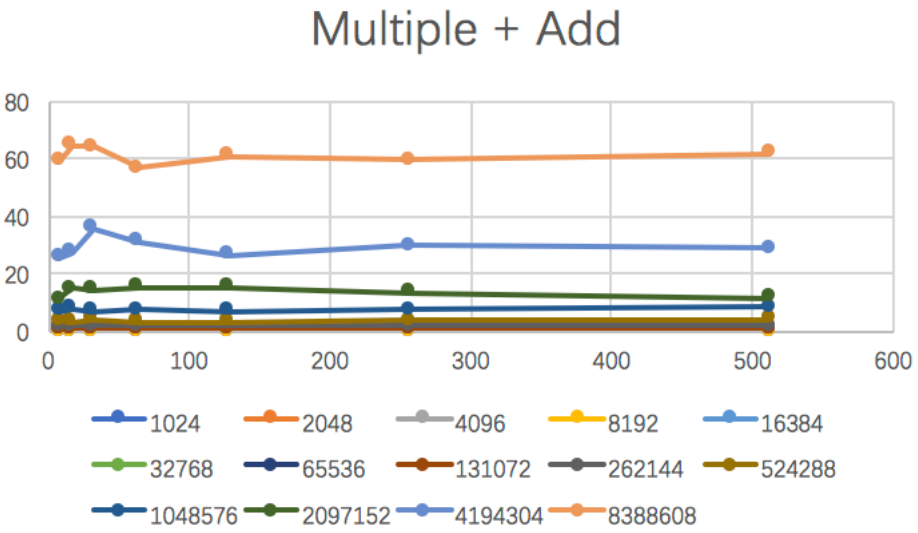
   Multiply:





Multiply + Add:



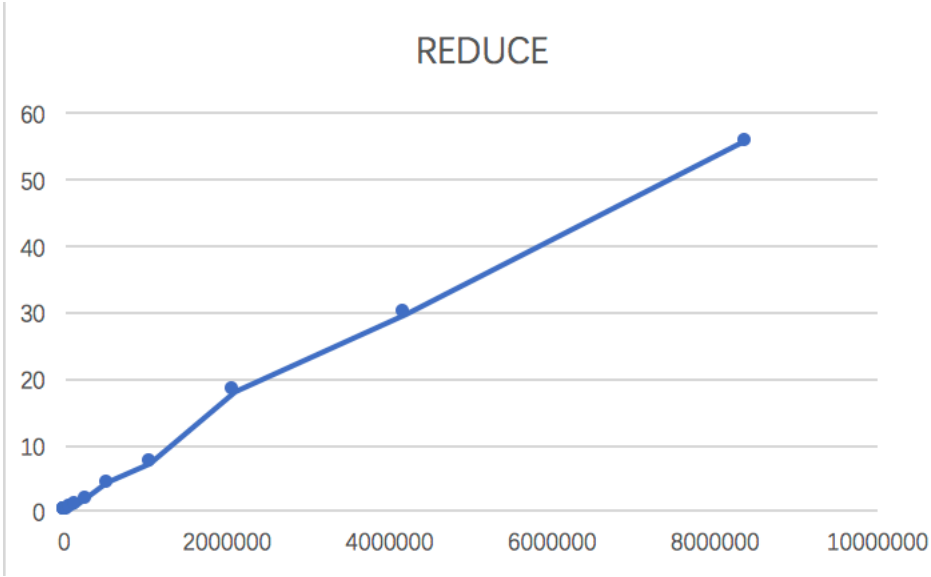


For Multiply and Multiply add we can see that in this experiment the key value which influence the performance a lot is Array Size. If we increase the Array Size we will get better performance.

And in perfect situation if we increase the work group size, the performance will increase. And at some point the performance will remain constant. What I get actually is that while doing little size computation, this concept is right. The PC condition impacts a lot when we do big size computation. The Intel GPU condition is affected by temperature and background process.

These graphs mean if we want to better use GPU, first we need to assign enough tasks to GPU and second we need to keep the running environment fresh, so that GPU can deal with tasks without interrupting. Also, proper work group size need to be selected

Reduction:



This graph tells us that double the size of array will also double the performance. I think this is because we haven't reached the limitation of this Intel GPU. If we make full use of GPU, the performance will remain constant. This is because we keep the work group size unchanged. And proper use of GPU parallel computing means assign enough jobs to the GPU.