CS 575

Project #6

Name: Chi Wen
ID: 933-276-677
Email: wench@oregonstate.edu

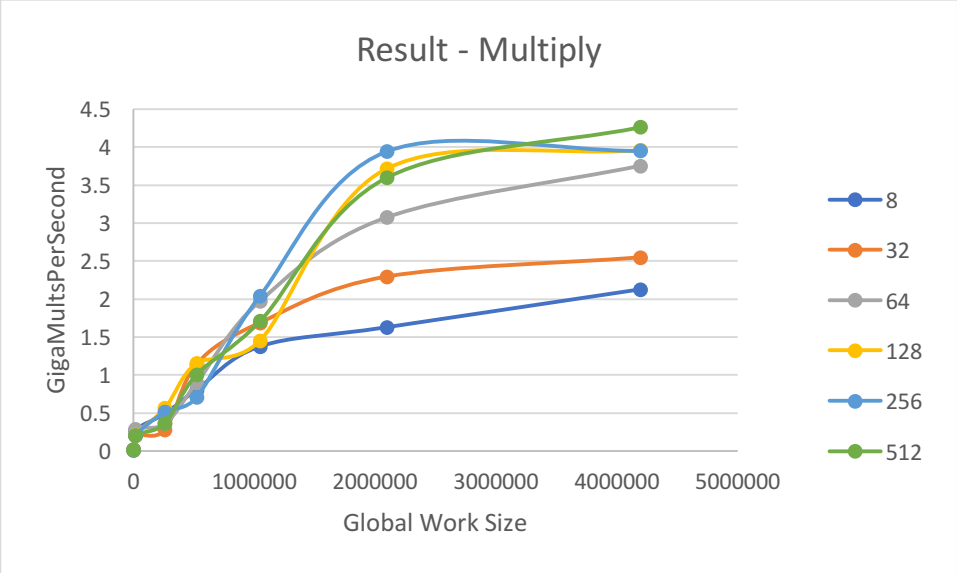
1. **What machine you ran this on?**
   I ran this on rabbit.

2. **Show the tables and graphs.**
   - Multiply

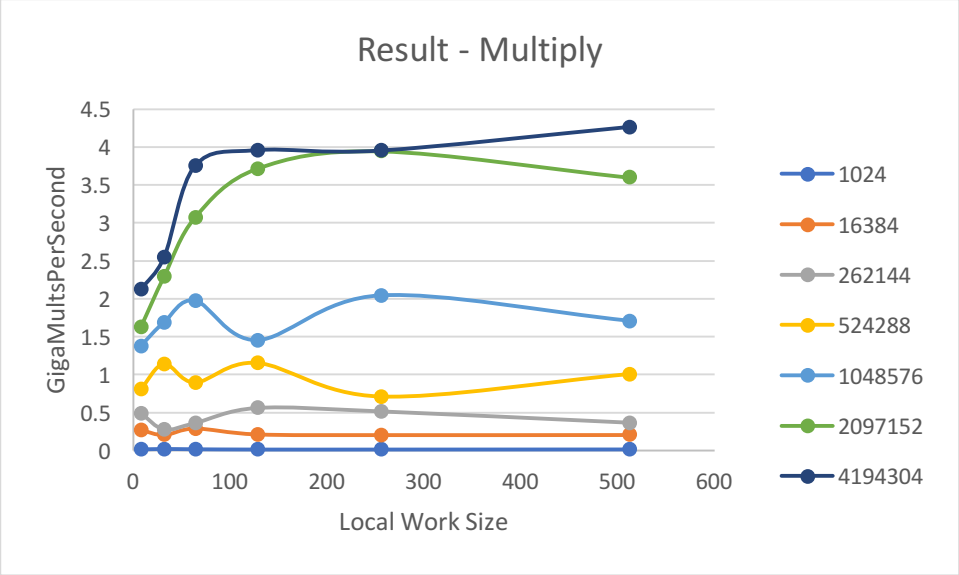
     The Global Work Size is set from 1024 to 4194304, and the Local Work Size is set from 8 to 512.

	1024	16384	262144	524288	1048576	2097152	4194304
8	0.014	0.271	0.492	0.805	1.375	1.627	2.124
32	0.019	0.206	0.275	1.142	1.687	2.294	2.547
64	0.016	0.29	0.36	0.895	1.972	3.073	3.751
128	0.013	0.214	0.562	1.155	1.452	3.714	3.957
256	0.013	0.205	0.516	0.708	2.042	3.944	3.952
512	0.014	0.206	0.365	1.007	1.707	3.598	4.262

Performance versus Global Work Size:



Performance versus Local Work Size:

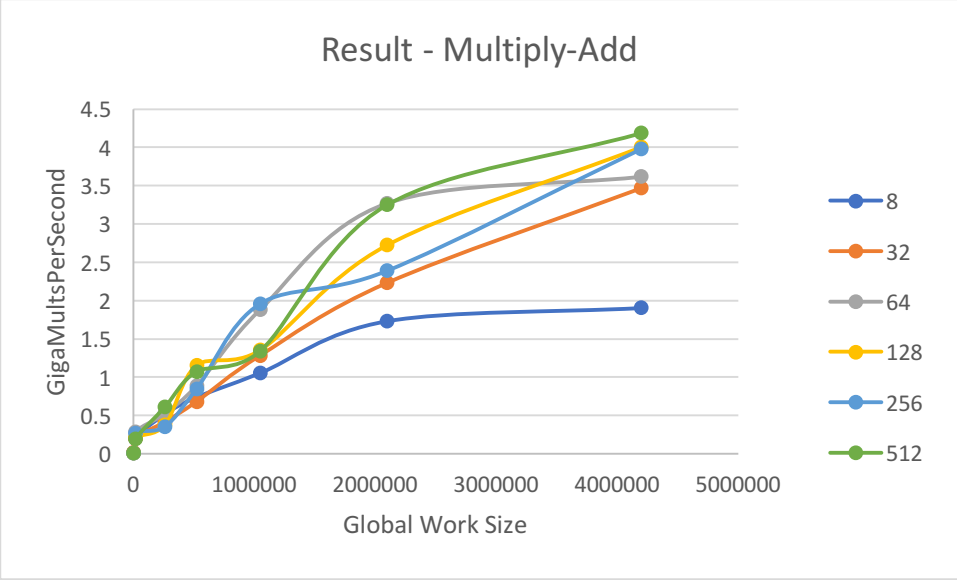


- Multiply-Add

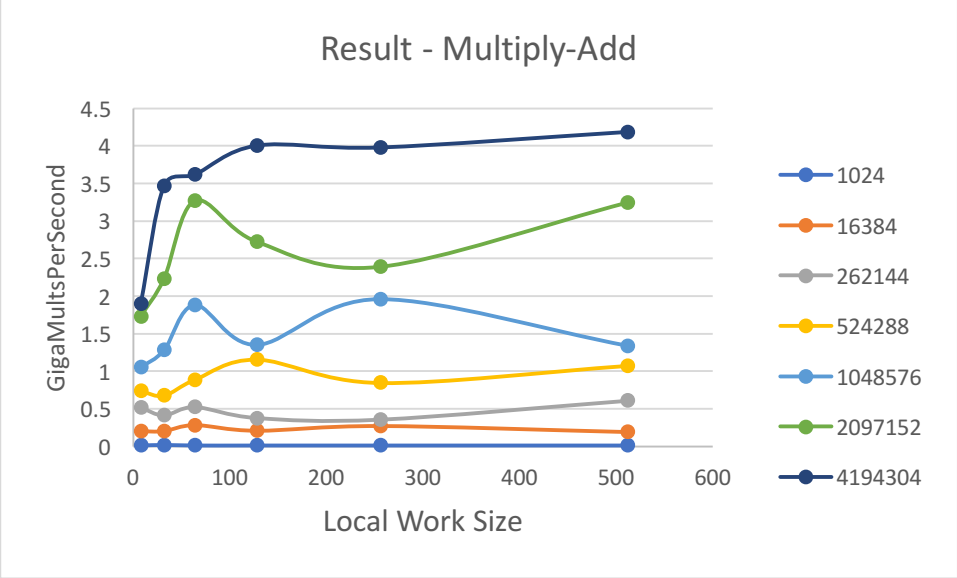
  The Global Work Size is set from 1024 to 4194304, and the Local Work Size is set from 8 to 512.

	1024	16384	262144	524288	1048576	2097152	4194304
8	0.013	0.203	0.517	0.738	1.053	1.73	1.902
32	0.018	0.205	0.413	0.68	1.281	2.23	3.466
64	0.013	0.281	0.524	0.889	1.88	3.27	3.617
128	0.013	0.21	0.377	1.155	1.353	2.724	4.003
256	0.013	0.27	0.355	0.847	1.959	2.39	3.978
512	0.013	0.193	0.61	1.073	1.34	3.248	4.185

Performance versus Global Work Size:



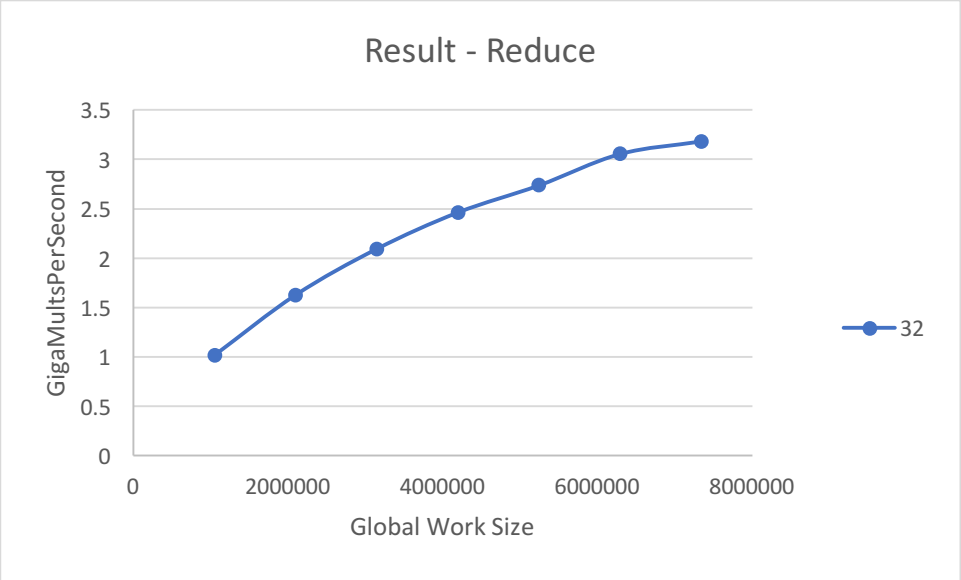
Performance versus Local Work Size:



- Multiply Reduction

  The Global Work Size is set from 1024 to 7340032, and the Local Work Size is 32.

	1048576	2097152	3145728	4194304	5242880	6291456	7340032
32	1.015	1.625	2.091	2.46	2.734	3.052	3.18



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

- Multiply and Multiply-Add
  Both Multiply and Multiply-Add's patterns are very similar to each other. In the "Performance versus Global Work Size" graph, it is oblivious that when the global work size increases, the performance increases. However, we can also see that when the local work sizes are small, even the global work size are increasing, the performance will increase very slowly. Moreover, even if the local work size is bigger,

once if the global work size is small, it won't make much changes to their performance.
In the "Performance versus Local Work Size" graph, we can observe that if the global work size isn't big enough, even increase the local work size, the performance is almost the same. And for the global work size which is 4194304, when increase the local work size, even it starts out increasing very fast, it slows down at one point, and remain almost the same performance.

- Multiply Reduction
  It starts off increasing very quickly, but in the end, it starts it level off a little.

4. **Why do you think the patterns look this way?**
   - Multiply and Multiply-Add
     Under a fixed local work size, if the global work size increase, that means there will be more work groups and there will be more work for computing unit to do, so there for, the performance increases.
     On the other hand, if the global work size is fixed, when the local work size is small, it will soon reach their limitations, and while the local work size is big, it will have better performance.

   - Multiply Reduction

Because each time it will reduce half of the size, and while the local work size is fixed to 32, the larger the global work size is, the better performance it will have.

5. **What is the performance difference between doing a Multiply and doing a Multiply-Add?**
   They are very similar to each other. However, Multiply has better performance than Multiply-Add, which is because Multiply-Add has extra operation that it need to compute, and it will make it slower than Multiply.

6. **What does that mean for the proper use of GPU parallel computing?**
   If there is a large size problem, it is suitable to use GPU parallel computing; however, if the problem is too small, GPU parallel computing is not a very good option. Also, having a right amount of local work size is very important, which we can see it shows clearly in the graph of “Performance versus Local Work Size”.