CS-575 Parallel Programming

Spring 2022

Name: Akhil Sai Chintala

Email: chintala@oregonstate.edu

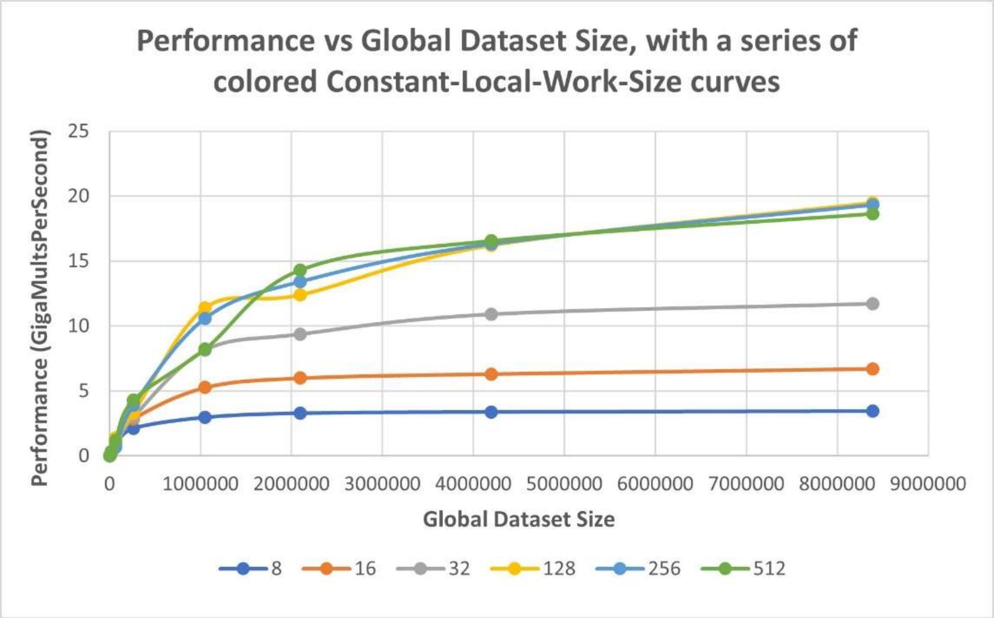
Project Name: #Project-6

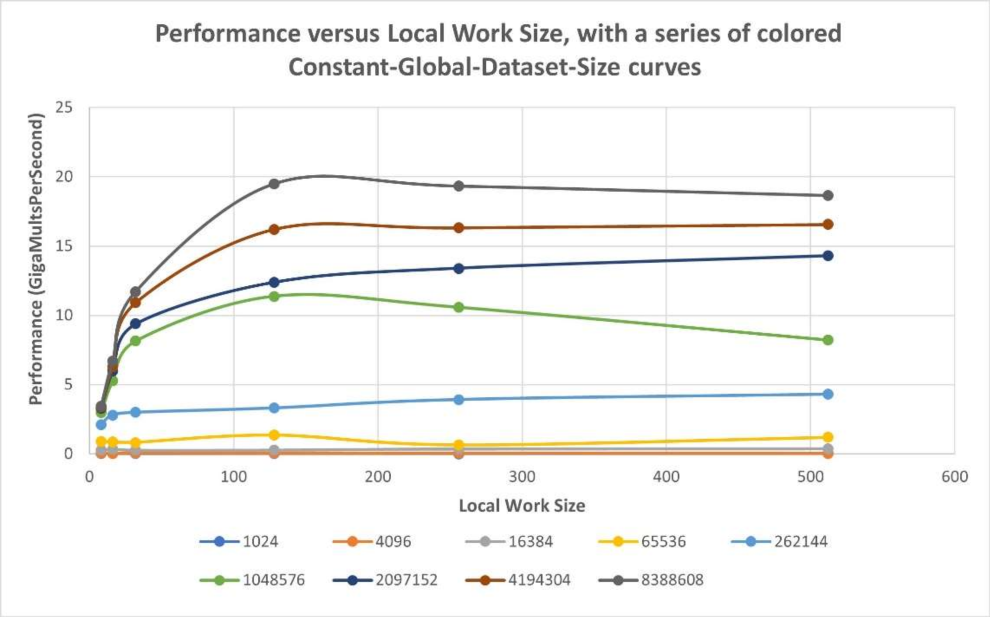
PART-1:

1. **What machine you ran this on?**
   I have executed the project-6 code on the rabbit server "rabbit.engr.oregonstate.edu".

2. **Show the tables and graphs**

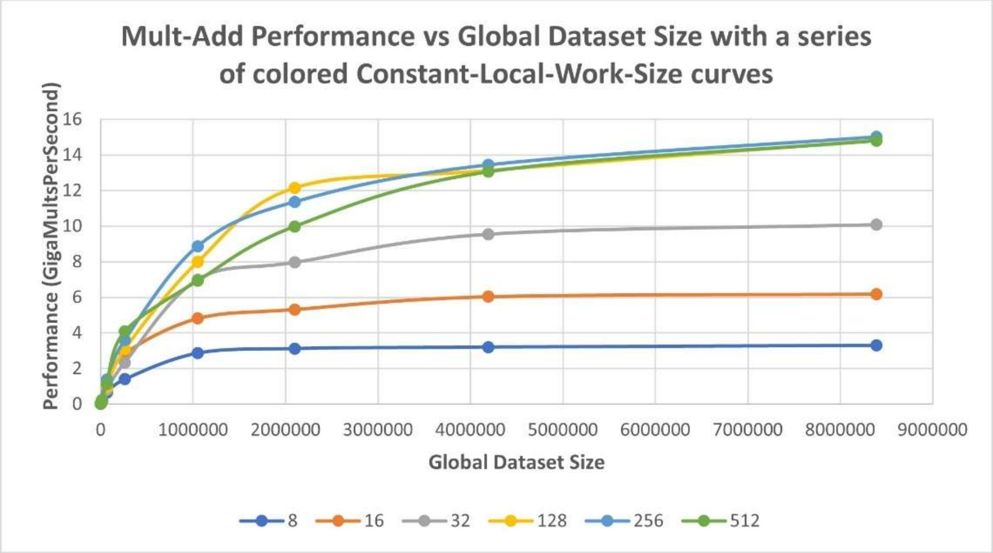
Global Dataset Size \ Local Size	8	16	32	128	256	512
1024	0.014	0.02	0.024	0.019	0.013	0.016
4096	0.064	0.037	0.067	0.046	0.055	0.043
16384	0.264	0.33	0.263	0.27	0.341	0.359
65536	0.906	0.865	0.844	1.374	0.645	1.195
262144	2.13	2.837	3.002	3.315	3.921	4.304
1048576	2.97	5.263	8.155	11.393	10.586	8.221
2097152	3.29	5.99	9.387	12.399	13.418	14.316
4194304	3.388	6.308	10.918	16.21	16.32	16.558
8388608	3.456	6.706	11.727	19.487	19.32	18.641



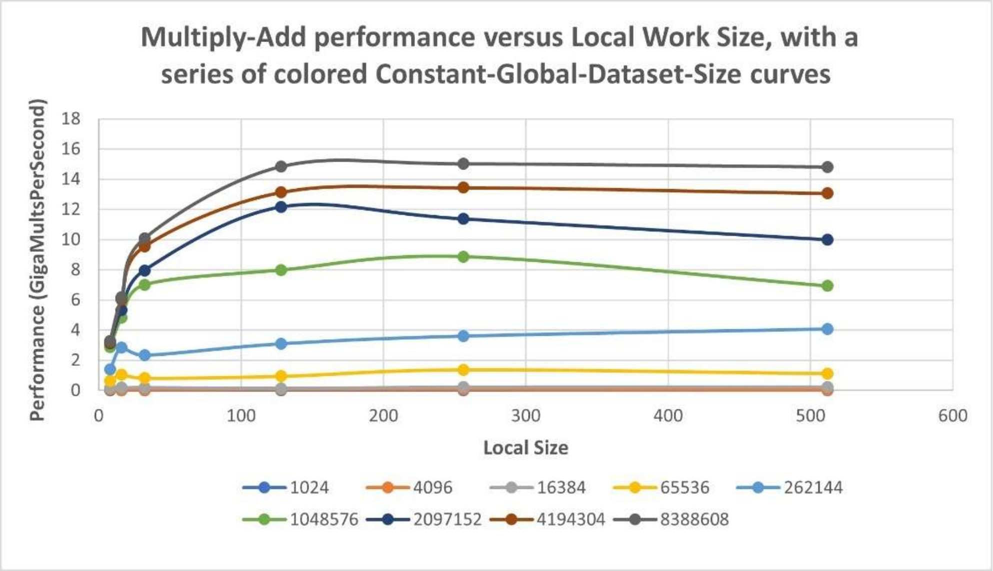


Performance versus Local Work Size, with a series of colored Constant-Global-Dataset-Size curves

Global Datasize\Local Size	8	16	32	128	256	512
1024	0.02	0.018	0.016	0.014	0.012	0.012
4096	0.062	0.056	0.058	0.061	0.062	0.049
16384	0.18	0.187	0.201	0.143	0.223	0.238
65536	0.658	1.037	0.821	0.945	1.377	1.128
262144	1.401	2.834	2.34	3.101	3.611	4.083
1048576	2.864	4.814	7.003	7.985	8.872	6.928
2097152	3.123	5.322	7.959	12.156	11.367	9.987
4194304	3.203	6.039	9.546	13.134	13.445	13.067
8388608	3.301	6.188	10.083	14.835	15.021	14.812



Mult-Add Performance vs Global Dataset Size with a series of colored Constant-Local-Work-Size curves



3. **What patterns are you seeing in the performance curves?**
   From the above graphs,
   a) **Performance Vs Global Dataset Size:** For the given global dataset size, the performance is increased when the local work size increases and it reaches maximum at 128 and remains constant till the end.
   b) Performance Vs Local Size: For the given values, the performance increases as the local size increases.

4. **Why do you think the patterns look this way?**
   When the global dataset size is very small, then the GPU memory will be free, and the threads will be waiting to process the data elements to compute them, because of this the work done is not enough on the GPU and cannot recover from the overhead.

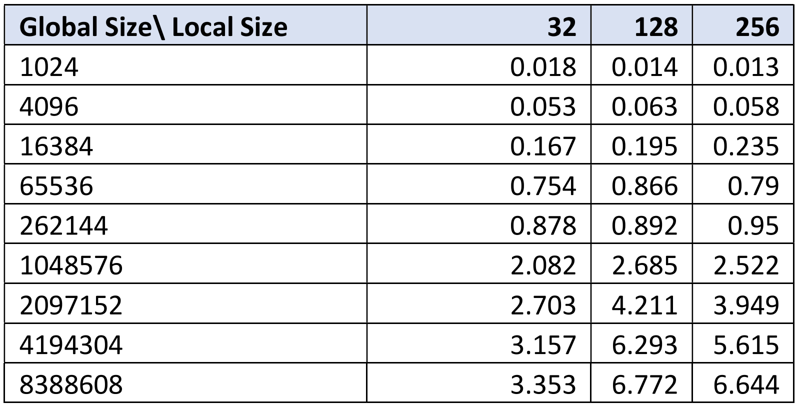
   When the local size is very small, then the number of processing elements will be more and waiting time is increased because of the idle time. Due to this unavailability of resources the compute time is wasted.

5. **What is the performance difference between doing a Multiply and doing a Multiply-Add?**
   According to the performance, multiply is better than the Multiply-Add. For multiplying arrays there's no wait time i.e., the processing time will be less. For multiplying-add, we must perform multiplication first and then addition which will take a bit longer i.e., more processing time.

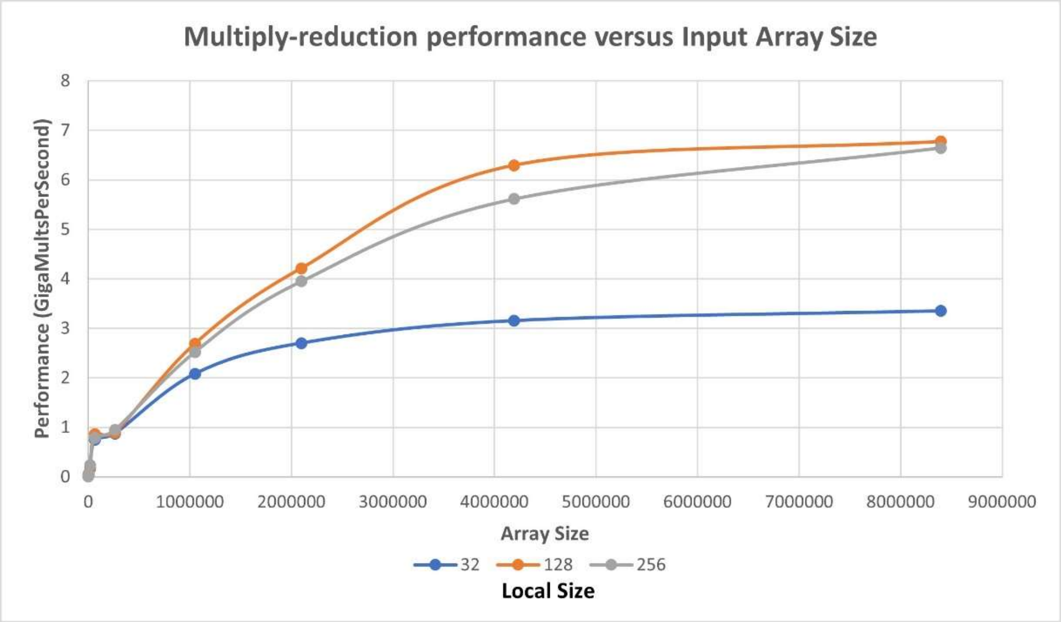
6. **What does that mean for the proper use of GPU parallel computing?**
   By looking at the performance shown in the graphs, block size 128 is the best option as it reaches maximum. Sizes more than 128 will have a constant performance and if the data size is less than 128 then performance will also be less (Based on graphs).

PART-2

1. **Show this table and graph**



Global Size\ Local Size	32	128	256
1024	0.018	0.014	0.013
4096	0.053	0.063	0.058
16384	0.167	0.195	0.235
65536	0.754	0.866	0.79
262144	0.878	0.892	0.95
1048576	2.082	2.685	2.522
2097152	2.703	4.211	3.949
4194304	3.157	6.293	5.615
8388608	3.353	6.772	6.644



2. **What pattern are you seeing in this performance curve?**
   From the above graph, the performance increases with increase in input array size and it reaches the maximum when the maximum value of array size is reached.

3. **Why do you think the pattern looks this way?**
   When the size of the array is less, then the GPU is not so busy i.e., it will be idle which results in less performance. If the array size is more then, the GPU is used effectively which results in better performance.

4. **What does that mean for the proper use of GPU parallel computing?**
   If the size of the data is less, then the proper use of GPU is not done because the idle time for computing the data will be more. As the data size increases, the GPU will be used more and the idle time will be decreased which results in more performance.