Name: Venkata Vasantha Phani Sai

Email:paramkuv@oregonstate.edu

Project: Project06

Onid: paramkuv

Class: CS\_575

1. What machine you run this on?

Ans. Rabbit.

2. Show the tables and graphs?

Ans.

|  |  |  |  |
| --- | --- | --- | --- |
| **GLOBAL\_SIZE** | **LOCAL\_SIZE** | **NUM\_WORK\_GROUPS** | **MegaMultiply Per Second** |
| 1024 | 8 | 128 | 0.013 |
| 1024 | 16 | 64 | 0.015 |
| 1024 | 32 | 32 | 0.016 |
| 1024 | 64 | 16 | 0.019 |
| 1024 | 128 | 8 | 0.014 |
| 1024 | 256 | 4 | 0.013 |
| 1024 | 512 | 2 | 0.018 |
| 2048 | 8 | 256 | 0.039 |
| 2048 | 16 | 128 | 0.025 |
| 2048 | 32 | 64 | 0.033 |
| 2048 | 64 | 32 | 0.037 |
| 2048 | 128 | 16 | 0.023 |
| 2048 | 256 | 8 | 0.046 |
| 2048 | 512 | 4 | 0.021 |
| 4096 | 8 | 512 | 0.055 |
| 4096 | 16 | 256 | 0.058 |
| 4096 | 32 | 128 | 0.052 |
| 4096 | 64 | 64 | 0.078 |
| 4096 | 128 | 32 | 0.06 |
| 4096 | 256 | 16 | 0.078 |
| 4096 | 512 | 8 | 0.049 |
| 8192 | 8 | 1024 | 0.221 |
| 8192 | 16 | 512 | 0.113 |
| 8192 | 32 | 256 | 0.142 |
| 8192 | 64 | 128 | 0.123 |
| 8192 | 128 | 64 | 0.123 |
| 8192 | 256 | 32 | 0.152 |
| 8192 | 512 | 16 | 0.141 |
| 16384 | 8 | 2048 | 0.266 |
| 16384 | 16 | 1024 | 0.275 |
| 16384 | 32 | 512 | 0.282 |
| 16384 | 64 | 256 | 0.295 |
| 16384 | 128 | 128 | 0.28 |
| 16384 | 256 | 64 | 0.248 |
| 16384 | 512 | 32 | 0.214 |
| 32768 | 8 | 4096 | 0.415 |
| 32768 | 16 | 2048 | 0.501 |
| 32768 | 32 | 1024 | 0.436 |
| 32768 | 64 | 512 | 0.397 |
| 32768 | 128 | 256 | 0.49 |
| 32768 | 256 | 128 | 0.473 |
| 32768 | 512 | 64 | 0.581 |
| 65536 | 8 | 8192 | 1.017 |
| 65536 | 16 | 4096 | 1.098 |
| 65536 | 32 | 2048 | 0.979 |
| 65536 | 64 | 1024 | 0.973 |
| 65536 | 128 | 512 | 0.742 |
| 65536 | 256 | 256 | 1.332 |
| 65536 | 512 | 128 | 0.807 |

|  |  |  |  |
| --- | --- | --- | --- |
| **GLOBAL\_SIZE** | **LOCAL\_SIZE** | **NUM\_WORK\_GROUPS** | **MegaMultiply-Adds Per Second** |
| 1024 | 8 | 128 | 0.021 |
| 1024 | 16 | 64 | 0.023 |
| 1024 | 32 | 32 | 0.019 |
| 1024 | 64 | 16 | 0.022 |
| 1024 | 128 | 8 | 0.024 |
| 1024 | 256 | 4 | 0.018 |
| 1024 | 512 | 2 | 0.019 |
| 2048 | 8 | 256 | 0.051 |
| 2048 | 16 | 128 | 0.044 |
| 2048 | 32 | 64 | 0.038 |
| 2048 | 64 | 32 | 0.028 |
| 2048 | 128 | 16 | 0.023 |
| 2048 | 256 | 8 | 0.039 |
| 2048 | 512 | 4 | 0.041 |
| 4096 | 8 | 512 | 0.096 |
| 4096 | 16 | 256 | 0.065 |
| 4096 | 32 | 128 | 0.065 |
| 4096 | 64 | 64 | 0.1 |
| 4096 | 128 | 32 | 0.095 |
| 4096 | 256 | 16 | 0.072 |
| 4096 | 512 | 8 | 0.046 |
| 8192 | 8 | 1024 | 0.126 |
| 8192 | 16 | 512 | 0.057 |
| 8192 | 32 | 256 | 0.187 |
| 8192 | 64 | 128 | 0.125 |
| 8192 | 128 | 64 | 0.148 |
| 8192 | 256 | 32 | 0.144 |
| 8192 | 512 | 16 | 0.182 |
| 16384 | 8 | 2048 | 0.224 |
| 16384 | 16 | 1024 | 0.329 |
| 16384 | 32 | 512 | 0.306 |
| 16384 | 64 | 256 | 0.359 |
| 16384 | 128 | 128 | 0.289 |
| 16384 | 256 | 64 | 0.266 |
| 16384 | 512 | 32 | 0.313 |
| 32768 | 8 | 4096 | 0.646 |
| 32768 | 16 | 2048 | 0.692 |
| 32768 | 32 | 1024 | 0.642 |
| 32768 | 64 | 512 | 0.746 |
| 32768 | 128 | 256 | 0.501 |
| 32768 | 256 | 128 | 0.579 |
| 32768 | 512 | 64 | 0.605 |
| 65536 | 8 | 8192 | 1.366 |
| 65536 | 16 | 4096 | 1.631 |
| 65536 | 32 | 2048 | 1.406 |
| 65536 | 64 | 1024 | 1.172 |
| 65536 | 128 | 512 | 0.89 |
| 65536 | 256 | 256 | 1.231 |
| 65536 | 512 | 128 | 1.506 |

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

Ans. Both parts of the graphs have a raise in performance by an increase in the size of the global and local dataset size. In short, the higher the size of the global and local dataset size the performance would be better according to this graph.

4. Why do you think the patterns look this way?

Ans.

Global Dataset Size: Performance increases with the increase in the size of the Global dataset because if the Global dataset is larger then that means that the maximum utilization of the GPU Parallelism computing the benefits of the OpenCL. If the Global dataset size is very small, then the benefits of parallel computing cant are utilized due to the higher overhead costs.

Local Dataset size: Dataset Size: Performance increases with the increase in the size of the Local dataset because each processing element will share the memory and work with the threads in the group so the graph will show the increase in performance.

5. What is the performance difference between doing a Multiply and doing a Multiply-Add?

Ans.

When we compare both graphs of Multiply and Multiply-add both graphs show a similar kind of performance except for the graph in the multiplication is very slightly performing well. It means that both multiplication and multiplication-add use similar FMA instructions.

6. What does that mean for the proper use of GPU parallel computing?

Ans.

The proper use of GPU parallel computing using the FMA instruction in the OpenCL to boost the performance of the multiplication and addition operations. This can be accomplished by using larger Global and Local Datasets sizes.

Part II

1. Show this table and graph?

Ans.

|  |  |  |  |
| --- | --- | --- | --- |
| **GLOBAL\_SIZE** | **LOCAL\_SIZE** | **NUM\_WORK\_GROUPS** | **MegaMultiply-reduction performance Per Second** |
| 1024 | 32 | 32 | 0.02 |
| 1024 | 64 | 16 | 0.022 |
| 1024 | 128 | 8 | 0.02 |
| 1024 | 256 | 4 | 0.027 |
| 2048 | 8 | 256 | 0.042 |
| 2048 | 16 | 128 | 0.035 |
| 2048 | 32 | 64 | 0.043 |
| 2048 | 64 | 32 | 0.039 |
| 2048 | 128 | 16 | 0.065 |
| 2048 | 256 | 8 | 0.04 |
| 4096 | 32 | 128 | 0.097 |
| 4096 | 64 | 64 | 0.097 |
| 4096 | 128 | 32 | 0.071 |
| 4096 | 256 | 16 | 0.1 |
| 8192 | 32 | 256 | 0.172 |
| 8192 | 64 | 128 | 0.176 |
| 8192 | 128 | 64 | 0.118 |
| 8192 | 256 | 32 | 0.207 |
| 16384 | 32 | 512 | 0.364 |
| 16384 | 64 | 256 | 0.384 |
| 16384 | 128 | 128 | 0.324 |
| 16384 | 256 | 64 | 0.326 |
| 32768 | 32 | 1024 | 0.603 |
| 32768 | 64 | 512 | 0.964 |
| 32768 | 128 | 256 | 0.833 |
| 32768 | 256 | 128 | 0.575 |
| 65536 | 32 | 2048 | 1.541 |
| 65536 | 64 | 1024 | 1.528 |
| 65536 | 128 | 512 | 1.743 |
| 65536 | 256 | 256 | 1.957 |
| 131072 | 32 | 4096 | 2.944 |
| 131072 | 64 | 2048 | 2.928 |
| 131072 | 128 | 1024 | 2.131 |
| 131072 | 256 | 512 | 3.734 |

2. What pattern are you seeing in this performance curve?

Ans.

As the Global dataset size and Local dataset size increases the performance of the graphs are increasing but in some case like in 64 and 128 input array size of the graph the performance is a bit unusual, I feel this is due to the load on the rabbit system.

3. Why do you think the pattern looks this way?

Ans.

Similar to what I explained in the part-1 4th answer as the global dataset and local datasets increase the performance of the graphs will be increased due to the utilization of GPU parallel computing.

But in the case of local dataset sizes 64, and 128 the performance is unusual I feel this is due to the load on the rabbit system.

4. What does that mean for the proper use of GPU parallel computing?

Ans.

By utilizing of the reduction to sum the workgroup instances will result in the smaller arrays in the global memory compared to the original global memory size. This results in better performance because the kernel can boost the calculation of the total sum significantly faster.