**CS 475/575 -- Spring Quarter 2022**

**Project #6**

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

Bhanu Prasanth Konda (934403560) kondab@oregonstate.edu

1. What machine you ran this on

I ran it on the rabbit server, which contains a Nvidia driver for GPU with the specs as follows.

Name = 'NVIDIA CUDA'

Vendor = 'NVIDIA Corporation'

Version = 'OpenCL 3.0 CUDA 11.4.158'

Profile = 'FULL\_PROFILE'

Number of Devices = 1

Device #0:

Type = 0x0004 = CL\_DEVICE\_TYPE\_GPU

Device Vendor ID = 0x10de (NVIDIA)

Device Maximum Compute Units = 15

Device Maximum Work Item Dimensions = 3

Device Maximum Work Item Sizes = 1024 x 1024 x 64

Device Maximum Work Group Size = 1024

Device Maximum Clock Frequency = 1071 MHz

1. Show the tables and graphs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset Size \ Local Size** | **8** | **16** | **32** | **128** | **256** | **512** |
| **1024** | 0.014 | 0.014 | 0.009 | 0.012 | 0.017 | 0.012 |
| **4096** | 0.064 | 0.07 | 0.061 | 0.086 | 0.083 | 0.065 |
| **16384** | 0.281 | 0.371 | 0.26 | 0.192 | 0.225 | 0.257 |
| **65536** | 0.721 | 0.8 | 0.832 | 0.94 | 0.739 | 1.087 |
| **262144** | 1.683 | 2.487 | 2.782 | 3.214 | 2.819 | 1.81 |
| **1048576** | 2.568 | 4.289 | 5.71 | 8.665 | 7.055 | 8.284 |
| **2097152** | 2.615 | 4.655 | 8.025 | 10.364 | 10.916 | 9.861 |
| **4194304** | 2.833 | 5.334 | 8.892 | 13.922 | 13.594 | 13.933 |
| **8388608** | 2.852 | 5.55 | 9.84 | 15.324 | 15.135 | 13.819 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset Size \ Local Size** | **8** | **16** | **32** | **128** | **256** | **512** |
| **1024** | 0.013 | 0.014 | 0.015 | 0.017 | 0.018 | 0.014 |
| **4096** | 0.078 | 0.067 | 0.057 | 0.063 | 0.068 | 0.051 |
| **16384** | 0.266 | 0.259 | 0.232 | 0.228 | 0.201 | 0.309 |
| **65536** | 0.708 | 0.847 | 0.907 | 0.845 | 0.748 | 0.761 |
| **262144** | 1.731 | 2.424 | 2.408 | 3.924 | 4.252 | 3.895 |
| **1048576** | 2.719 | 4.247 | 6.472 | 9.276 | 9.298 | 9.419 |
| **2097152** | 2.77 | 5.185 | 8.242 | 13.182 | 13.487 | 12.553 |
| **4194304** | 2.934 | 5.605 | 9.948 | 16.011 | 17.462 | 16.913 |
| **8388608** | 3.022 | 5.856 | 10.447 | 19.002 | 19.068 | 18.058 |

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

In the above performance vs Data Set graphs, performance lines increase till 128 and remain constant on top of them

In the performance vs local size graph, performance increase with the increase in the local size.

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

When there is a small data set then there will be a lot of gpu memory and threads waiting to process the data so the computation is faster as the size increases the performance can be observed accordingly.

Now, when the local block size is less that means there is more data needs to be processed and there is a lot of wait for each block to complete. Hence the graph appears in that way like each like is stacked over each other.

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

During the multiplying arrays, there is no wait or a process that needs to be done after the multiplication but during the multiplication and addition, we have to multiply and add a number which will take this process a bit more time to complete.

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

Based on the graphs, 128 block size is the best option. When we observe the graph, even if we take a block size of more than 128, it settles at the same performance and when we take a lower block size, the performance decreases.

Part 2:

1. Show this table and graph

|  |  |  |  |
| --- | --- | --- | --- |
| **DatasetSize\Local size** | **32** | **128** | **256** |
| **1024** | 0.011 | 0.011 | 0.013 |
| **4096** | 0.039 | 0.051 | 0.053 |
| **16384** | 0.199 | 0.185 | 0.139 |
| **65536** | 0.803 | 0.643 | 0.789 |
| **262144** | 0.76 | 0.669 | 0.747 |
| **1048576** | 1.806 | 2.177 | 2.287 |
| **2097152** | 2.262 | 3.619 | 3.952 |
| **4194304** | 2.9 | 5.128 | 4.675 |
| **8388608** | 3.541 | 6.685 | 6.415 |

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

From the above graph, we can observe that the performance increases with an increase in the local block size

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

When the dataset size is less that there is a lot of GPU remaining idle so the performance is less and with the increase in the dataset the GPU is more efficiently used.

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

Proper use of GPU is when we have a large dataset and a lot of independent processes to do. If there is a small dataset then it is better to use the processor to do the processing.