Naveen Tumkur Ramesh Babu
Computer Science and Engineering, OSU
Class CSE5441- 12:45-2:05 batch

LAB REPORT FOR PROGRAM-4

Lab4p1, lab4p2 are output file

PART 1) Matrix Multiplication Program.

-bash-4.2$ ./lab4p1

****

CUDA compute parameters are:

GPU with 1 grid,with 1024 blocks and with 1024 threads per block

Size of the matrix: 1024 x 1024

Time taken by the serial version:6.853352 seconds

Time taken by the CUDA version:0.011702 seconds

****

Checking that the two resulting matrices are equivalent.

The two resulting matrices are equivalent

The compute Structure used for my program are:-

Input Matrix size: 1024 X 1024

1 Grid, 1024 blocks and 1024 threads per block.

My program has 1024 (threads per block) * 1024(blocks per grid) * 1(grid) = 1024 * 1024 = Matrix size

Hence the number of threads spawned would be equal to total number of elements in the matrix(1024*1024).

GFlops

Matrix Size : 1024 X 1024

Operations performed(Both CUDA and Serial version) = 1024 * 1024 * 1024 * 2 = 2.147483648 * (10^9)

GFlops for serial version is : (2.147483648 * (10^9))/(6.853352 * 10^9)

= 0.31334793

GFlops for CUDA Version is : (2.147483648 * (10^9))/(0.011702 * 10^9)

= 183.51

Resulting Performance

The matrix size of my program : 1024 X 1024

Time taken by serial program : 6.853352 seconds

Time taken by CUDA Version : 0.011702 seconds

Therefore, I observed significant performance improvement with the usage of GPU for computation. The serial program took nearly 7 sec but with the help of GPU I was able to compute in 0.011 sec.

PART 2) Sobel Edge Detection Program.

```
-bash-4.2$ ./lab4p2 environment.bmp serial_img.bmp cuda_img.bmp
Image Info ::
        Height=4160 Width=3120
********************************************************************************
Serial and  CUDA Sobel edge detection.

Input image: nature.bmp    (Height is: 4160 pixels, Width is : 3120 pixels)
Serial output image is:      serial_img.bmp
CUDA output image is:      cuda_img.bmp

CUDA computation specifications are:
Grids =  1 grids
Blocks=  64 blocks
tpb= 1024 threads per block

********************************************************************************
Performing serial Sobel edge detection.
Performing CUDA parallel Sobel edge detection.
********************************************************************************
Time taken for serial Sobel edge detection: 27.369054
Convergence Threshold: 115

Time taken for CUDA Parallel Sobel edge detection: 5.565523
Convergence Threshold: 115
********************************************************************************
-bash-4.2$
```

The compute Structure used for my program are:-

Input Matrix size: 4160(height) X 3120(Width)

1 Grids, 64 blocks and 1024 threads per block.

My program has 1024 (threads per block) * 64(blocks per grid) * 1(grid) = 1024 * 64 = 65,536.

	Serial	CUDA
Coins.bmp 246(height) X 300(Width)	Time taken for serial Sobel edge detection: 0.277660 Convergence Threshold: 50	Time taken for CUDA Parallel Sobel edge detection: 0.117289 Convergence Threshold: 50
Nature.bmp 4160(height) X 3120(Width)	Time taken for serial Sobel edge detection: 27.369054 Convergence Threshold: 115	Time taken for CUDA Parallel Sobel edge detection: 5.565523 Convergence Threshold: 115
Fall_img.bmp 4160(height) X 3120(Width	Time taken for serial Sobel edge detection: 42.920459	Time taken for CUDA Parallel Sobel edge detection: 9.460759

	Convergence Threshold: 176	Convergence Threshold: 176

Resulting Performance Improvement

Yes, I have seen a significant performance improvement when using GPU for calculation. For example, Nature.bmp took 27.36 seconds for serial version while it took 5.56 seconds for CUDA version with multiple threads and GPU. The performance improvement is nearly 5 times.

NOTE:-

1) **CudaMemCpy** takes time to copy contents from Host to device and vice-versa. So, programs with small input might not give very good results. I have checked with two large images of 39Mb size to get to know the benefits of using a GPU.
2)Time used to calculate is chrono.
3) I have used double type for magnitude calculation. This will give more precision in result.
4) Performance exponentially increases compared to serial version as we increase size of image.