ECEN 5593: ADVANCED COMPUTER ARCHITECTURE

FINAL PROJECT PROPOSAL

SANJANA KALYANAPPAGOL
saka2821@colorado.edu

SPRING 2017

MOTIVATION:

Matrix Multiplication Research and Comparison

Utilizing graphics hardware for general purpose numerical computations has become a topic of considerable interest. The implementation of streaming algorithms, typified by highly parallel computations with little reuse of input data, has been widely explored on GPUs. The matrix multiplication is the key operation for many computationally intensive algorithms. It is a kernel operation used in many transform, image and discrete signal processing application as well as robotic applications. Matrix mathematics applies to several branches of science, as well as different mathematical disciplines. It is also used in the field of Computer Graphics. Nowadays, we can observe the results of matrix mathematics in every computer-generated image that has a reflection, or distortion effects such as light passing through rippling water, etc. Before computer graphics, the science of optics used matrix mathematics to account for reflection and for refraction and It also helps calculate the electrical properties of a circuit, with voltage, amperage and resistance. Because of its trending applications, new algorithms and new techniques are being developed on configurable devices too. Its regular data access pattern and highly parallel computational requirements suggest matrix-matrix multiplication as an obvious candidate for efficient evaluation on GPUs but, surprisingly near optimal GPU implementations are pronouncedly less efficient than current cache-aware CPU approaches

Image Processing Algorithms Implementation and Analysis

Many image processing algorithms require dozens of floating point computations per pixel, which can result in slow runtime even for the fastest of CPUs. The slow speed of a CPU is a huge bottleneck to productivity. While increased pixel counts lead to greater area coverage and higher resolution, it also results in higher image processing time. So the implementation method to be designed will significantly improve performance of image processing algorithms and real time image and video processing, the only overhead will be the data transfer times.

OBJECTIVES OF THE PROJECT:

Matrix Multiplication**:**

Matrix Multiplication is a heavy computation algorithm used in many applications as mentioned above which involves concepts like CUDA, shared memory, threads, blocks and memory accessing. The aim is to implement concepts like Tiling, CGMA and Shared Memory to analyse the performance of Matrix Multiplication in GPU vs CPU. As an extension to the this project, I would also like to show the comparison while implemented using OpenCL.

Image Processing Algorithm:

I am interested to get exposed to implementing a few image processing algorithms using CUDA. The project will be primarily being focused on basic Image processing algorithms. The objective is to implement Image-Invert Algorithm and Grayscale algorithm for 32 bits per pixel (bpp) color encoded bitmap images and to measure execution times on CPU as compared to

GPU using NVIDIA Jetson GPU Processor. The project also aims to analyse the performance of the algorithms in different processors and present a detailed report regarding the same.

METHODOLOGY:

Implementation for both:

- Understanding Matrix Multiplication and Image Processing Algorithms.
- Implementing the algorithm in C to obtain performance analysis on CPU.
- Execute the code by porting to CUDA and getting it to run on GPU.
- Profiling for timing analysis

Matrix Multiplication:

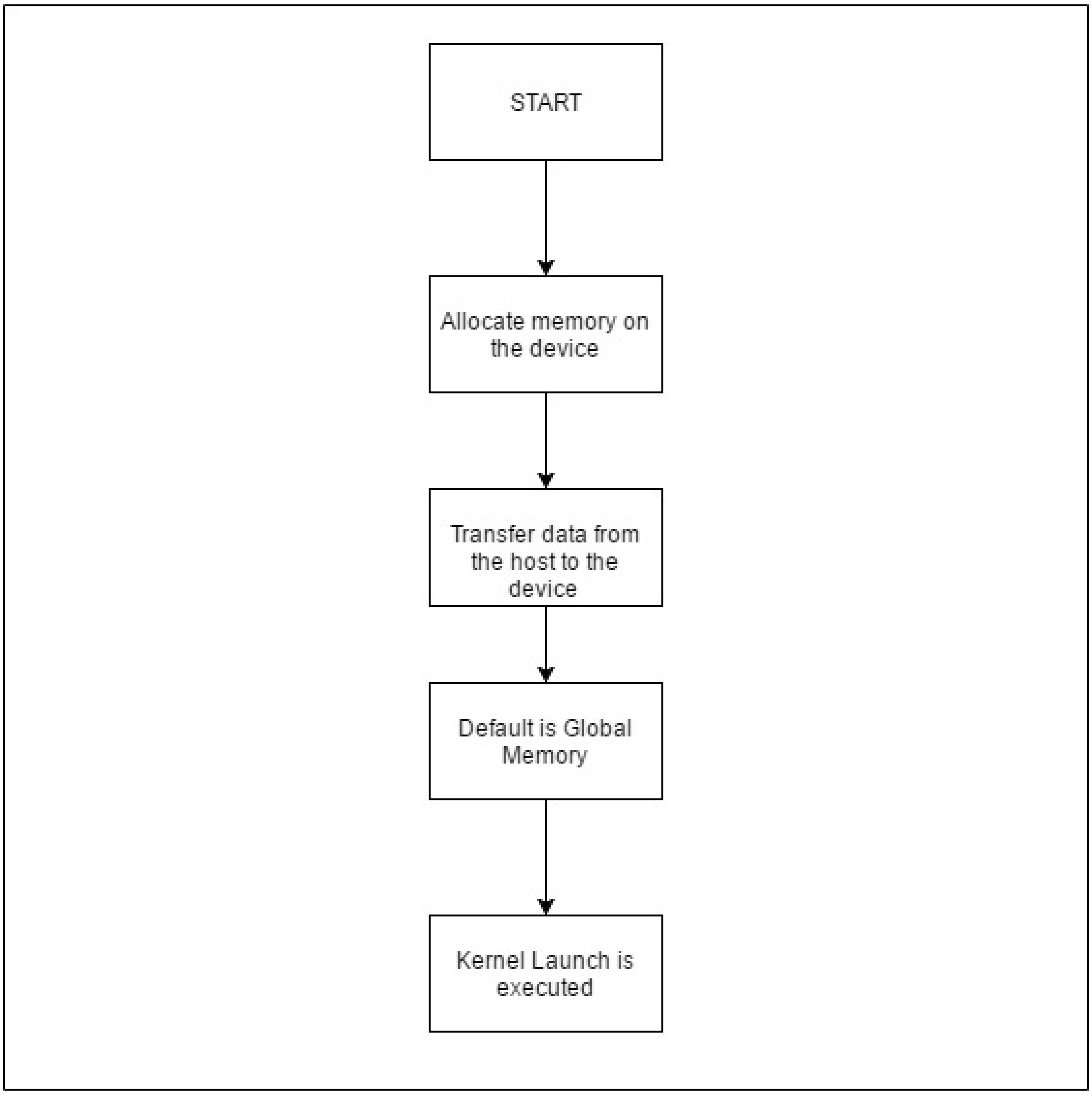
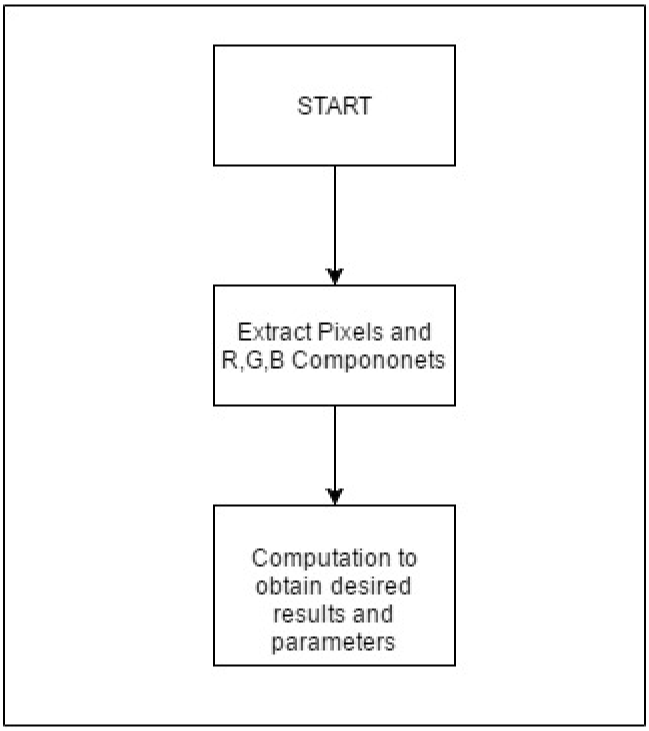

START
Allocate memory on the device
Transfer data from the host to the device
Default is Global Memory
Kernel Launch is executed

Image Processing:


START
Extract Pixels and R,G,B Compononets
Computation to obtain desired results and parameters

PROJECT OUTCOMES AND GOALS:

The project involves implementation of the algorithms such as Matrix Multiplication and Image Processing Algorithms on CUDA. It gives an exposure and experience in implementation of algorithms on GPU as well as CPU. At the end of the project, I will be able to analyse the performance and execution times of algorithms on different processors. The goal is to provide a detailed report of performance and timing analysis of algorithms which are trending in current day technology. This should be able to point out differences in performance variance of different algorithms on GPU vs CPU.

REFERENCES:

- https://graphics.stanford.edu/papers/gpumatrixmult/gpumatrixmult.pdf
- https://people.eecs.berkeley.edu/~sangjin/2013/02/12/CPU-GPU-comparison.html
- http://www.nvidia.com/object/cuda-home.html
- http://www.nvidia.com/content/nvision2008/tech_presentations/Game_Developer_Track/NVISION08-Image_Processing_and_Video_with_CUDA.pdf
- http://supercomputingblog.com/cuda/advanced-image-processing-with-cuda