**EE 5123**

**Computer Architecture (Spring 2018)**

GPU Benchmarking using CUDA framework (Nvidia)

BY

GOPAL VISHWAKARMA

2018

**CONTENTS**

1. **Introduction** 
   1. What are Graphics Processing Units
   2. What is CUDA Programming Model
   3. What is HPP (Heterogeneous parallel programming)
2. **Approach and Execution of the project**
3. **Results**
4. **Findings/Conclusions**

**References**

**Aim of The Project**

The project evaluates performance of CPU & GPU and time taken to execute a code on each processor. In this, we will be giving huge array of numbers and compute it on CPU as well as on GPU. And we can test the time required to process it, also we will be testing this benchmarking program on different GPU since we have two PC that has different versions of Nvidia GPU.

**Introduction**

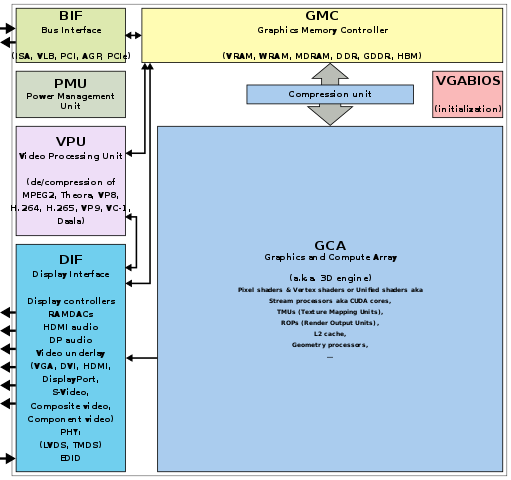
**1.1 What are Graphics Processing Units?**

A Graphics Processing Unit (GPU) is a microprocessor that has been designed specifically for the processing of 3D graphics. The processor is built with integrated transform, lighting, triangle setup/clipping, and rendering engines, capable of handling millions of math-intensive processes per second. GPUs form the heart of modern graphics cards, relieving the CPU (central processing units) of much of the graphics processing load. GPUs allow products such as desktop PCs, portable computers, and game consoles to process real-time 3D graphics that only a few years ago were only available on high-end workstations.

Used primarily for 3-D applications, a graphics processing unit is a single-chip processor that creates lighting effects and transforms objects every time a 3-D scene is redrawn. These are mathematically-intensive tasks, which otherwise, would put quite a strain on the CPU. Lifting this burden from the CPU frees up cycles that can be used for other jobs.

However, the GPU is not just for playing 3D-intense videogames or for those who create graphics (sometimes referred to as graphics rendering or content-creation) but is a crucial component that is critical to the PC's overall system speed. In order to fully appreciate the graphics card's role, it must first be understood.

Many synonyms exist for Graphics Processing Unit in which the popular one being the graphics card. It’s also known as a video card, video accelerator, video adapter, video board, graphics accelerator, or graphics adapter.

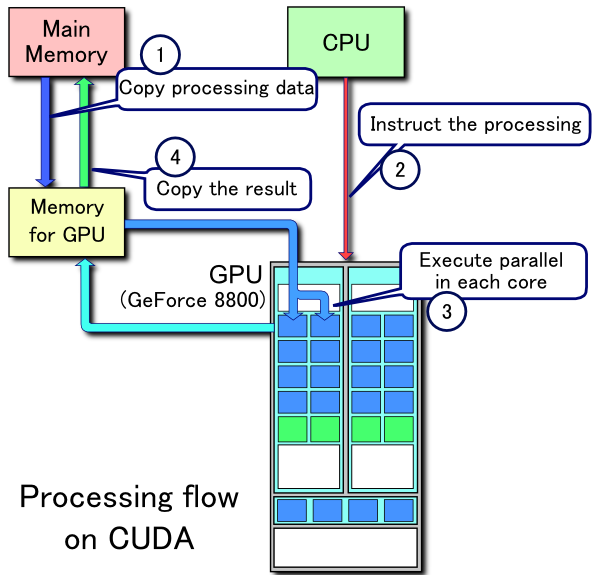


***Fig.1 Components of a GPU***

**1.2 What is CUDA Programming Model?**

CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing – an approach termed GPGPU (General-Purpose computing on Graphics Processing Units). The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels.

The CUDA platform is designed to work with programming languages such as [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), and [Fortran](https://en.wikipedia.org/wiki/Fortran). This accessibility makes it easier for specialists in parallel programming to use GPU resources, in contrast to prior APIs like [Direct3D](https://en.wikipedia.org/wiki/Direct3D) and [OpenGL](https://en.wikipedia.org/wiki/OpenGL), which required advanced skills in graphics programming. Also, CUDA supports programming frameworks such as OpenACC and OpenCL.



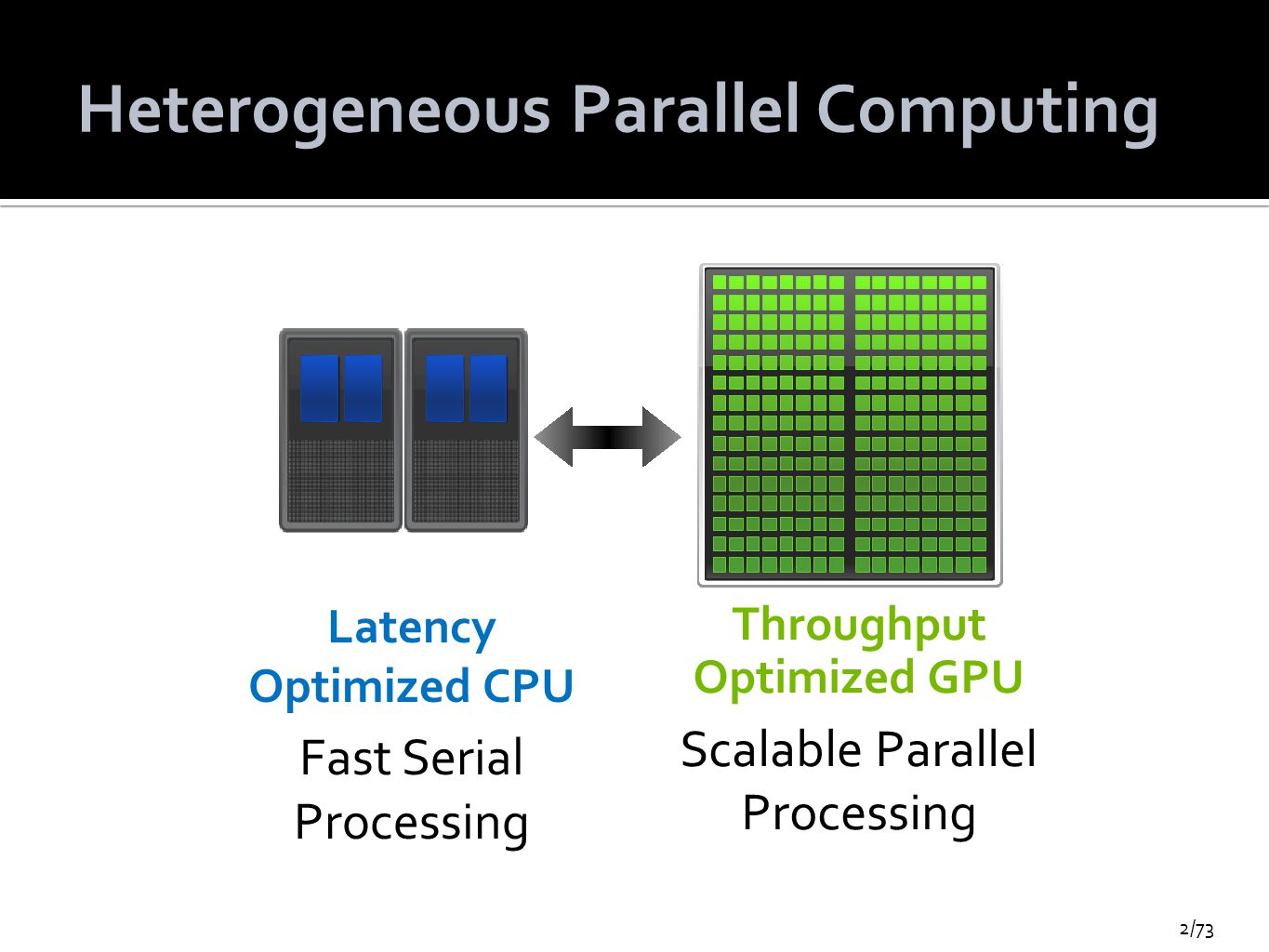
***Fig.2 Simple Processing Flow of CUDA***

**1.3 What is HPP (Heterogeneous parallel programming)?**

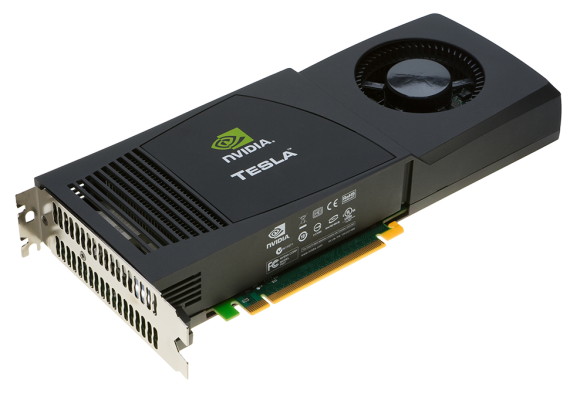
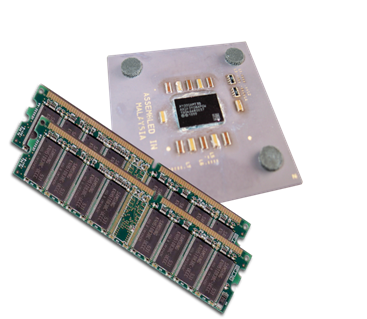
Heterogeneous Parallel Programming is about making the Graphics Card Processor (GPU) help CPU to do the heavy jobs in comparatively very much less time, for which the CPU, working alone might have taken much more time.

In essence, it boosts system performance dramatically i.e. exponentially.

Usually GPU’s are made up of hundreds of core nearly same as we use quad core (4 cores) or dual core (2 cores) in PC. Generally, inside PC, operating system creates software threads that ismanaged by CPUscheduling algorithm. However, GPU’s have hundreds of powerful cores and which are hardware threads that can be controlled by CPU. Previously, GPU are made for only for graphical application like PC games, animation, movies. But Nvidia added their hardware kit inside GPU so that we can use those cores for general computation.



***Fig.3 Heterogeneous parallel computing***

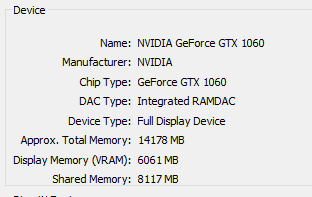


***Fig.4 The CPU and its memory (host memory) Fig.5 The GPU and its memory (device memory)***

**Approach of the project**

The project is divided into three segments they are as follow,

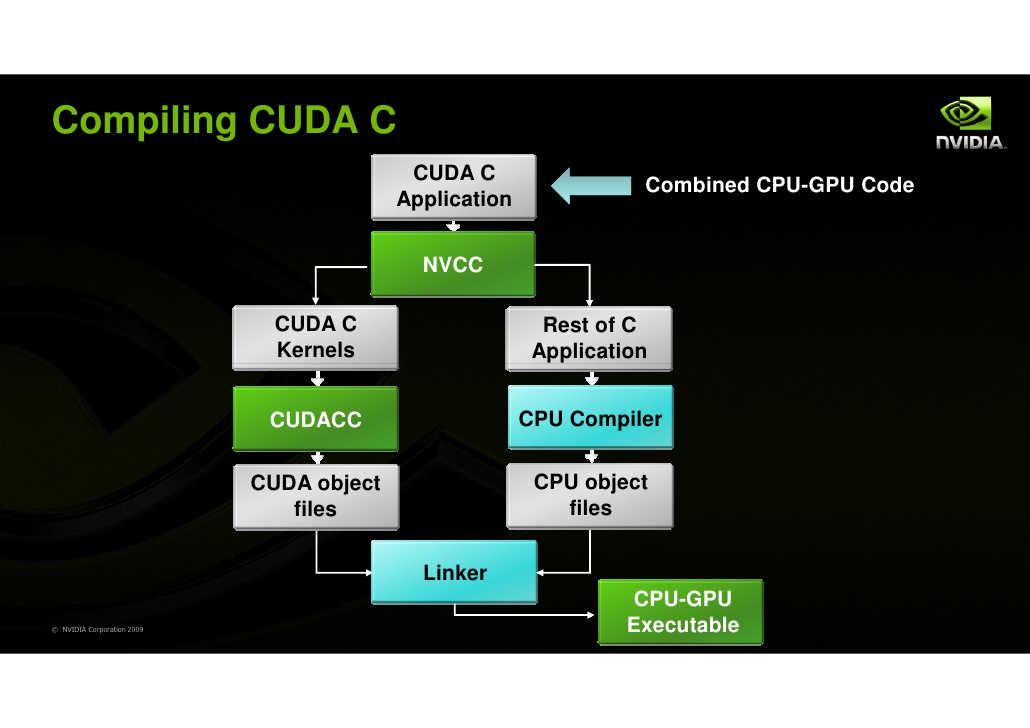
1. Getting GPU device properties using CUDA API’s.
2. Addition of a vector array (CPU as well as GPU).
3. Multiplication of 2D array.

****

***Fig.6 GPU details***

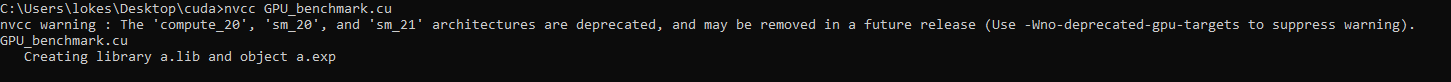
**Execution of the Project**

Firstly, the program needs to be compiled in the Nvidia CUDA compiler using the command,  
After that the program creates a .exe file which is ready for use.  
Nvcc (Nvidia CUDA compiler)



***Fig.7 Compiler***

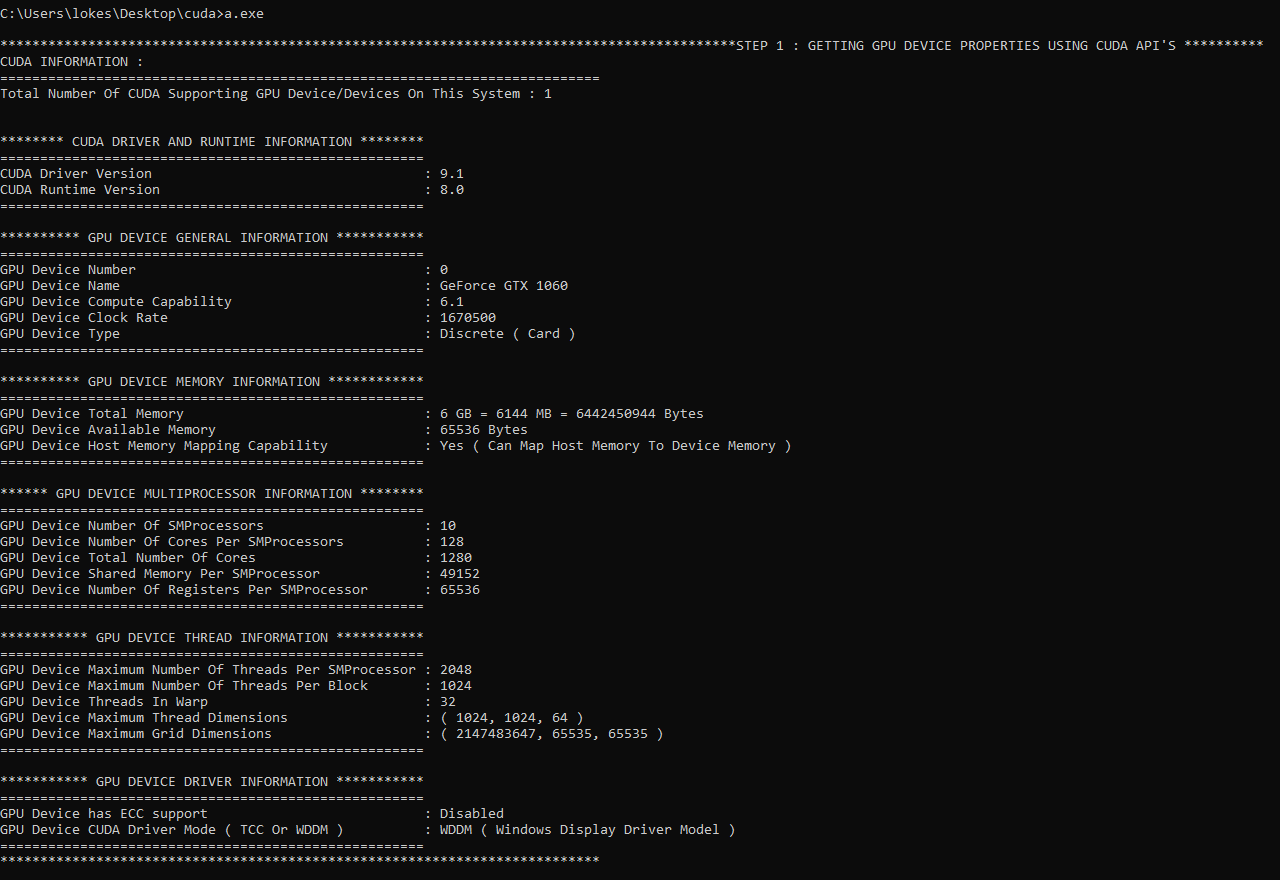
**Results**

1. Compilation of the program

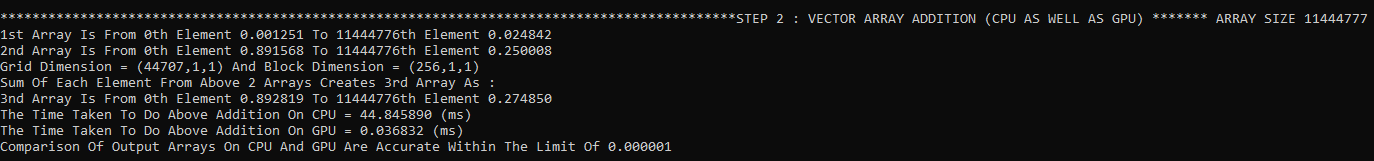
“a.exe” executable file is generated after compilation process.

Execution of code is as below:

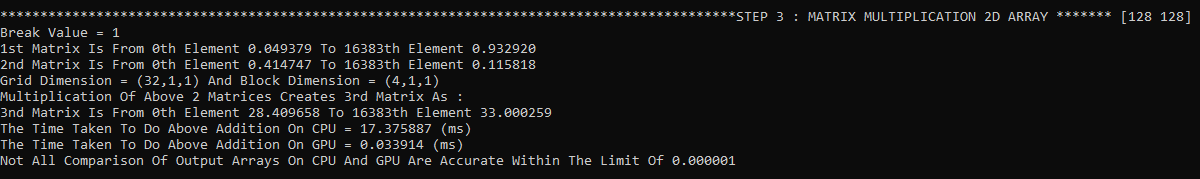
STEP 1: Getting GPU device properties using CUDA API



STEP 2: Addition of vector array



STEP 3: Multiplication of the 2D array



Time taken by GPU and CPU for different code snippets are given below,

Vector addition of 2 arrays ( Array size 11444777 values)



2D array multiplication of two arrays( Each array of 128 rows, 128 columns)



**Findings/Conclusions**

Performance of CPU Vs multiple GPU’s is evaluated, and time taken to execute is recorded. In this project, addition of two huge array of numbers (approx. 11million values) and multiplication two 2D array each has 128 rows and 128 columns was given as input and computed on CPU(Intel i7 processor) as well as on GPU and the results were recorded, also tested the time required to process it. Also, benchmark program is tested on different computers which has Nvidia GPU to compare the results.

**References**

[1] https://en.wikipedia.org/wiki/Graphics\_processing\_unit.

[2] https://en.wikipedia.org/wiki/CUDA.

[3] www.nvidia.com/docs/IO/116711/sc11-cuda-c-basics.pdf.

[4] https://web.stevens.edu/sss/summer-2010/7-7-2010-CUDA\_Programming.

[5] www.cs.kent.edu/~wcheng/GA%20works*.*

[6] www.glearning.tju.edu.cn.

[7] Computer Architecture In-Class Slides.