International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.1, Issue.2, pp-358-363 ISSN: 2249-6645

# An Introduction to Graphical Processing Unit

Jayshree Ghorpade[1], Jitendra Parande[2], Rohan Kasat[3], Amit Anand[4]
[1](Department of Computer Engineering, MITCOE, Pune University) India
[2](SunGard Global Technologies, Pune) India
[3](Department of Computer Engineering, MITCOE, Pune University) India
[4](Department of Computer Engineering, MITCOE, Pune University) India

## ABSTRACT

**Today's world requires maximum computing speed. The progress that the CPU has achieved over the past 2 decades, though tremendous, has now reached a point of stagnation. To overcome this, a new highly parallel and multithreading processor optimized for high degree of computation was introduced, which was named as the Graphics Processing Unit (GPU) by NVIDIA or the Visual Processing Unit (VPU). A Graphics Processing Unit (GPU) is a single-chip processor primarily used to manage and boost the performance of video and graphics. This paper talks about the reasons for choosing GPU to accelerate the computation. This paper also states where GPU will work more efficiently than the CPU.**

***Keywords*** **– CPU, data parallelism, GFLOPS, GPU, SPMD**

## I. INTRODUCTION

CPU frequency growth is now limited by physical matters and high power consumption. Their performance is often raised by increasing the number of cores. Present day processors may contain up to four cores (further growth will not be fast), and they are designed for common applications, they use MIMD (multiple instructions / multiple data). Each core works independently of the others, executing various instructions for various processes

The GPU is a specialized processor efficient at manipulating and displaying computer graphics. The term was defined and popularized by Nvidia as "a single chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engines that is capable of processing a minimum 10 million per seconds"[1]. There are mathematically-intensive tasks, complex algorithms which would put quite a strain on the CPU. GPU lifts this burden from

the CPU and frees up cycles that can be used for other jobs. The highly parallel graphics processing unit (GPU) is rapidly gaining maturity as a powerful engine for computationally demanding applications. GPU hides latency with computation not with cache! The GPU's performance and potential offer a great deal of promise for future computing systems. One of the most important challenges for GPU computing is to connect with the mainstream fields of processor architecture and programming systems, as well as learn from the parallel computing experts of the past.

The GPU is a chip that functions on the same principle as the CPU with the one important difference that it has nothing to do with any part of the system that is not part of the graphics package on the computer. GPU is essentially a CPU that is specifically designed and dedicated to the control of graphics. The end result is an easier to control graphics package and better response time based on computer commands. Games with intensive graphics end up running a lot quicker and multimedia that you find at online websites tend to be a lot better as well. The advantages of having a GPU are therefore quite easy to notice from those outcomes and that is why people are now clamoring to have GPU devices installed into their computers.

When we compare GPUs with CPUs over the last decade in terms of Floating point operations (FLOPs), we see that GPUs appear to be far ahead of the CPUs as shown in Fig.1.

GPUs came into existence with only image and graphics computation in mind. But now GPUs has evolved into an extremely flexible and powerful processor in terms of

- Programmability
- Precision
- Performance

So GPUs are well suited for fast, efficient, non graphical computing too.

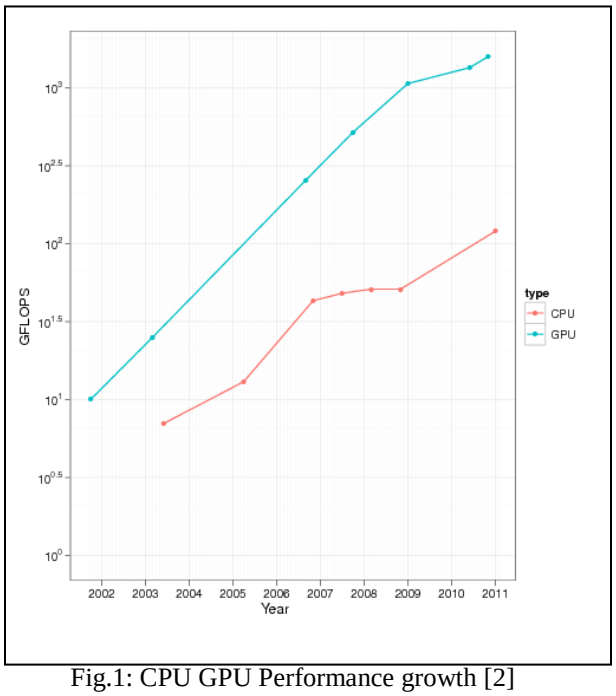


Fig.1: CPU GPU Performance growth [2]

## II. CPU GPU COMPARISON

The CPU or Central Processing Unit is where all the program instructions are executed in order to derive the necessary data. The advancement in modern day CPUs have allowed it to crunch more numbers than ever before, but the advancement in software technology meant that CPUs are still trying to catch up. A Graphics Processing Unit or GPU is meant to alleviate the load of the CPU by handling all the advanced computations necessary to project the final display on the monitor.

Originally, CPUs handle all of the computations and instructions in the whole computer, thus the use of the word 'central'. But as technology progressed, it became more advantageous to take out some of the responsibilities from the CPU and have it performed by other microprocessors.

The GPU is a device that is beneficial primarily to people that has intensive graphical functions on their computer. In other words, if you just use Microsoft Office and the e-mail page of your browser when you are on the computer, chances are very good that the GPU will not add that much to your computing experience. However, if you play video games and look at videos on the internet on a frequent basis, what you will discover is that installing a GPU onto your computer will greatly improve the performance you get out of the entire thing. Improving computer performance is always a good thing and this is why the GPU has become very popular in recent years. GPU computing is on the rise and continuing to grow in popularity and that makes the future very friendly for it indeed.

In GPU computing the CPU calculations are replaced by Graphics Processing Units. Migrating large scale algorithms and entire kernel onto the GPU co-processors help in arriving at the answer much faster and thus decreases the processing time. GPUs are never a completed replacement for CPUs but complementary. Parallel operation of CPU and GPU has found to increase the performance. CPUs offload the tasks which are better performed by GPU leading to high performance computing. GPU excel CPUs in certain computational tasks.

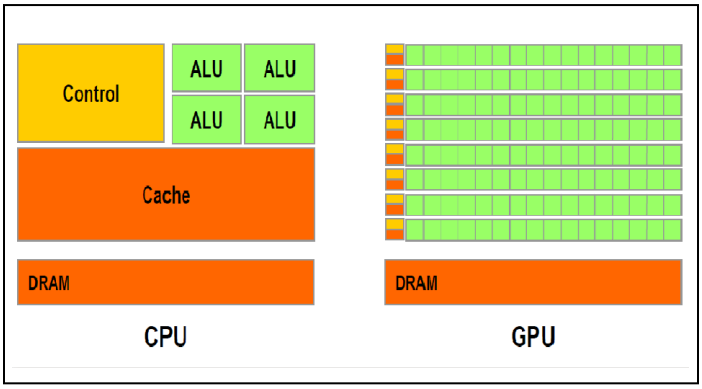


Fig.2: CPU GPU Comparison [3]

Whether it is CPU or GPU every processing unit has its own memory (cache) and shared memory (DRAM). Since it is very hard to transfer data between these structures one should avoid using complex data structures and messaging in their parallel algorithms. As it can be seen in Fig. 2, GPU has many ALU (Arithmetic Logic Unit) as compared to the CPU. So, it is able to perform multiple instructions execution at the same time. This provides the GPU with a high degree of parallelism which aids efficient computation. But it lacks the cache space. Owing to these structure differences we can deduce that an algorithm implemented for parallel structures may still work with better performance on a multi-core CPU when it could not be efficiently parallelized on GPU.

## III. WORKING OF GPU

The programmable units of the GPU follow a single program multiple-data (SPMD) programming model. For efficiency, the GPU processes many elements (vertices or fragments) in parallel using the same program. Each element is independent from the other

elements, and in the base programming model, elements cannot communicate with each other. All GPU programs must be structured in this way: Many parallel elements each processed in parallel by a single program. Each element can operate on 32-bit integer or floating point data with a reasonably complete general-purpose instruction set. Elements can read data from a shared global memory and, with the newest GPUs, also write back to arbitrary locations in shared global memory.

A graphics processing unit is a dedicated graphics rendering device for a personal computer, workstation, or game console. Modern GPUs are very efficient at manipulating and displaying computer graphics. But it is also used in general purpose computation in various computation intensive algorithms. These algorithms can harness the high multiplicity of the GPU to improve their performance. Mapping the general purpose computations on the GPU is very much similar to manipulating the computer graphics. GPU computing applications are structured in the following way [4]:

1. The programmer directly defines the computation domain of interest as a structured grid of threads.
2. An SPMD general-purpose program computes the value of each thread.
3. The value for each thread is computed by a combination of math operations and both read accesses from and write accesses to global memory. Unlike in the previous two methods, the same buffer can be used for both reading and writing, allowing more flexible algorithms (for example, in-place algorithms that use less memory).
4. The resulting buffer in global memory can then be used as an input in future computation.

The GPU is organized as multiple SIMD (Single instruction, multiple data) groups. Within one SIMD group, all the processing elements execute the same instruction in synchronization. A set of threads that execute in this way is called a "warp". Branching is allowed, but if threads within a single warp follow different execution paths, there may be some performance loss.

Memory interfaces are wide and achieve highest bandwidth when that access width is fully utilized. For applications that are memory bound, this means that all threads in a warp should access adjacent data elements when possible. For example, neighboring threads in a warp should access neighboring elements in an array. This may require some rearrangement of data layout or data access patterns.

The GPU offers multiple memory spaces that can be used to exploit common data-access patterns: in addition to the global memory, there are constant memory (read-only, cached), texture memory (read-only, cached, optimized for neighboring regions of an array), and per-block shared memory (a fast memory space within each warp processor, managed explicitly by the programmer).

## IV. CPU GPU WORK SHARING

For efficient use of the GPU one must find areas in the execution path of the program to send to the GPU, instead of offloading the entire code to GPU. This leads to better resource utilization. In short we must use the CPU for operations involving memory references and logical statements, while the computation intensive part of the code must be sent to the GPU.

Since the GPU is a coprocessor on a separate PCI-Express card, data must first be explicitly copied from the system memory to the memory on the GPU board.

As shown in Fig.3, the CPU has an input data stream, from where it receives the data to be processed. It has a global memory through which it references the tasks to be performed. Whenever a specific task is selected to be sent to the GPU, the CPU checks for an available unit of the GPU to which it can assign the task. On completion of the task the GPU signal the CPU and processing resumes. Since the GPU has multiple such units capable of a high degree of computation, parallelism is achieved and computation speeds up.

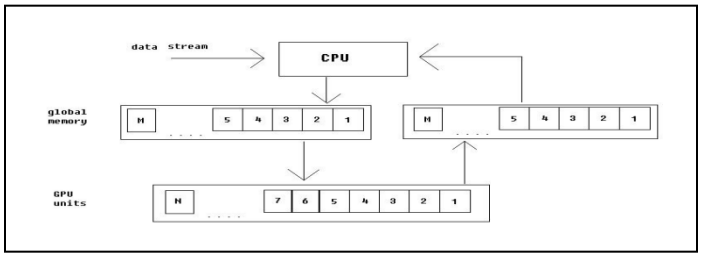


Fig.3: Data transfer between GPU and CPU

## V. BENEFITS AND LIMITATIONS

### *Benefits:*

1. Application that require large number of parallel threads work well.
2. Allows use of per-block shared memory.
3. Allows use of "data parallelism" in applications.
4. Easier to calculate reciprocal and reciprocal square root.
5. Can perform large amount of computation per data element.
6. If the synchronization is infrequent then GPU can handle them.

### *Limitations:*

1. Applications with limited concurrency do not fully utilize the potential of the GPU.
2. Even with many threads, if all the threads are doing different work then GPU is not utilized fully.
3. Frequent global synchronization requires an expensive global barrier.
4. If there is high degree of point-to-point synchronization among random threads the GPU does not work well.
5. Frequent communication between CPU and GPU hampers the performance of GPU.
6. Applications which require small amount of computation do not work well on GPU.

## VI. COMPARATIVE STUDY OF DIFFERENT GPU'S

There is always a constant and tough struggle for supremacy among the manufacturers of computer components like CPUs, graphics cards, system memory modules, coolers, etc. There is fierce competition in each price category, especially among top-end products. The graphics card market is a vivid example of that. Having the world's fastest graphics card under one's belt is not only prestigious but also profitable because it proves the manufacturer's technical superiority and promotes its sales in other price sectors.

Up to this moment the Nvidia GeForce GTX 580 has been the fastest single-GPU graphics card although AMD could offer its dual-processor Radeon HD 5970 as an alternative. On March 8, 2011, AMD released an even faster dual-GPU product, Radeon HD 6990. NVidia hasn't taken long to respond and has just rolled out its own dual-processor GeForce GTX 590.

### *Similarities:*

They do not differ much in terms of the peak power draw: 375 watts for the Radeon HD 6990 and 365 watts for the GeForce GTX 590. AMD recommends a 750W or higher power supply with two 150W power connectors for its graphics cards. A 1200W or higher power supply is recommended for a CrossFireX tandem built out of two Radeon HD 6990s. NVidia has the following recommendations: 700 and 1000-watt power supplies for a single GeForce GTX 590 and a SLI tandem, respectively. Each card has a single connector for building multi-GPU configurations. It is located in the top front part of the PCB.

### *Differences:*

The Radeon HD 6990 carries two full-featured Cayman GPUs. They are indeed full-featured because dual-processor cards used to be equipped with cut-down versions of GPUs in the past. The GPU frequency of the Radeon HD 6990 is only 50 MHz lower than that of the Radeon HD 6970 and equals 830 MHz. However, there is a high-speed mode you can trigger by means of the abovementioned switch near the CrossFireX connector. The card's GPU frequency is 880 MHz in that mode, but AMD says that turning that switch on will make your warranty void. The card's GPU frequency is lowered to 150 MHz in 2D applications to save power.

As opposed to AMD, NVidia equips its GPUs with such caps. The company didn't disable any subunits in the GPUs of its GeForce GTX 590, either. Each of the card's GPUs has 512 unified shader processors, 64 texture-mapping units and 48 raster operators. In other words, we've got two GeForce GTX 580 processors on a single PCB here. Their frequencies are lowered more than those of the AMD Radeon HD 6990, though. The GeForce GTX 590 clocks its GPUs at 607/1215 MHz, which is 21.4% lower than the clock rates of the GeForce GTX 580 (772/1544 MHz). The reason for this reduction is clear enough. If NVidia used the clock rates of the GTX 580 for the GTX 590, the latter's heat dissipation and power consumption would be beyond all reasonable limits. The GeForce GTX 590 drops its GPU clock rates to 51/101 MHz in 2D mode as a power-saving measure.

The GPU-Z tool is a tool that detects the CPU, RAM, motherboard chipset, and other hardware features of a modern personal computer, and presents the information in one window. It reports about the two cards as shown in fig4 and fig5.


Fig.4: ATI Radeon HD6990 Specifications [5]


Fig.5: NVidia GTX 590 Specifications [5]

The Radeon HD 6990 carries a total of 4 gigabytes of graphics memory (2 gigabytes per each GPU) whereas the GeForce GTX 590 has 1.5 gigabytes of onboard memory for each GPU or 3 gigabytes in total. As usual, AMD installs Hynix chips on its Cayman-based reference cards. These chips have a voltage of 1.5 volts and a rated frequency of 5000 MHz. The card's memory frequency is 5000 MHz, too, but is lowered to 600 MHz in 2D mode. The memory bus is 256 bits wide.

The GeForce GTX 590 comes with Samsung chips that have a rated access time of 0.4 nanoseconds and a rated frequency of 5000 MHz. However, the card clocks them at 3414 MHz only, which is 15% lower than the memory frequency of the GeForce GTX 580 and 10% lower than that of the GTX 570. The memory frequency is lowered to 270 MHz in 2D mode. The memory bus is 384 bits wide.

## VII. FUTURE WORK

GPU has gained over CPUs because they are more powerful and cheaper compared to CPUs. The main area of work in future for the GPU will be to reduce the cost and provide a huge multi-processing capability to the users all around. The future of GPU rests in making it as a co-processor which performs much of the computation for the CPU. This, as of now, is difficult to do since the languages and software tools available for combining GPU process with CPU are still in their preliminary stage of development. So, a major scope for future improvement is to develop new languages and software tools specifically to take advantage of high level of parallelism.

The other thing that can be improved with engineering is GPU communication with the CPU or the NIC (Network Interface Card). At present it takes a longer and slower route to copy contents from the CPU memory to the GPU blocks and vice-versa. This uses up a lot of time of computation. But with engineering we can develop better architectures to improve the communication speed of the GPU, reducing altogether the time required for computation.

## VIII. CONCLUSION

Thus we can conclude the following advantages of GPU over CPU such as

1. GPU's contain much larger number of dedicated ALU's then CPU.
2. GPU's also contain extensive support of stream processing paradigm. It is related to SIMD processing. [4]
3. Each processing unit on GPU contains local memory that improves data manipulation and reduces fetch time.

The Graphical Processing Unit is visually and visibly changing the course of general purpose computing. The future of the GPU certainly has much more promise on the horizon than the general-purpose CPU. Although the GPU will not overtake the CPU as the main processor, we do think that the GPU has much more potential for expanding our computing experience. The CPU has a large amount of logic dedicated to branch prediction, whereas stream processing does not require as much of this type of logic. The GPU is much better at parallelism than the CPU and as the gap in the transistor rate expansion continues to grow the GPU parallelism performance will also continue to grow. The power of solving these highly parallel problems has immense implications to the scientific community because we are the able to change the evolution of scientific computation from the CPU growth curve that double every 18 months to GPU growth curve that doubles about every 6 months.

## REFERENCES

[1] *Graphics Processing Unit – Wikipedia* (http://en.wikipedia.org/wiki/graphics_processing_unit)

[2] *CPU GPU Trends over time*. (http://csgillespie.wordpress.com/2011/01/25/cpu-and-gpu-trends-over-time/)

[3] *GPGPU: OPENCL vs CUDA vs ArBB* (http://www.keremcaliskan.com/gpgpu-opencl-vs-cuda-vs-arbb/)

[4] Peter Zalutski – "*CUDA–Supercomputing for masses*"

[5] Sergey Lepilov –"*Equilibrium: AMD Radeon HD 6990 vs. NVidia GeForce GTX 590*."