International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 1, December 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

GPU Computing: An Introduction

Matthew N. O. Sadiku, Adedamola A. Omotoso, Sarhan M. Musa

Roy G. Perry College of Engineering, Prairie View A&M University, Prairie View, Texas

ABSTRACT

The graphics processing unit (GPU) is a computer chip that performs rapid mathematical calculations. GPU is a ubiquitous device which appears in every computing systems such PC, laptop, desktop, and workstation. It is a many-core multithreaded multiprocessor that excels at both graphics and non-graphic applications. GPU computing is using a GPU as a co-processor to accelerate CPUs scientific and engineering computing. The paper provides a brief introduction to GPU computing.

KEYWORDS: *graphics processing unit, GPU computing, heterogeneous computing, hybrid computing*

How to cite this paper: Matthew N. O. Sadiku | Adedamola A. Omotoso | Sarhan M. Musa "GPU Computing: An Introduction" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-1, December 2019, pp.705-706, URL: www.ijtsrd.com/papers/ijtsrd29648.pdf

IJTSRD29648

INRODUCTION

The efficiency of any computer simulations depends on three factors [1]: the formulation of the theory describing the process, numerical methods employed, and the hardware capabilities. Graphics Processing Unit (GPU) computing is regarded today as the most powerful computational hardware. GPU is not the only type of accelerator core that has gained interest recently. Other include field programmable gate array (FPGA) and the Cell Broadband Engine (Cell BE), which receive less attention [2].

Originally, GPU was conceived and developed for rendering graphics. However, due to it high performance and low cost, it has become the new standard of non-graphic application such as image processing, image restoration, filtering, interpolation, and reconstruction. It has become an indispensable part of today's computing systems. In recent years, substantial efforts were made to adapt many algorithms for massively-parallel GPU-based systems since the GPU can perform many calculations simultaneously.

GPU computing is the application of a GPU to do general purpose scientific and engineering computing. Central Processing Units (CPUs) are task-parallel processors, while GPUs are data-parallel. GPU computing is not replacing CPU computing. Each approach has its own advantages and limitations. A GPU is used along with a CPU to accelerate scientific and engineering applications. For this reason, CPU-GPU computing is also known as heterogeneous computing or hybrid computing.

GPU BASICS

The term GPU was popularized by NVIDIA Corporation in 1999 when the company introduced the first GPU. NVIDIA's CUDA architecture (code named Fermi) is the first architecture to deliver all of the features required for highly demanding HPC applications. The features include high level of double-precision floating-point performance, ECC protection from the registers to DRAM, and support for languages including C, C++, FORTRAN, Java, MATLAB, and Python. Fermi is the first complete architecture for GPU computing [3]. Besides NVIDIA, other GPU vendors include Intel, ATI, Sony, and IBM.

GPU is specialized for compute-intensive, highly data parallel computation, which what graphics rendering is all about. Although GPU can be used for 2D data, it is essential for rendering of 3D animations and video. GPU has a unique design of 'many-core' architecture, and each core is able to carry out thousands of threads simultaneously. The memory of GPU consists of a large number of cache blocks, and each block can be independently accessed [4].

Before GPU was invented, graphics on a personal computer were performed by a video graphics array (VGA) controller. The GPU is designed for a class of applications with the following characteristics [5]:

- Computational requirements are large
- Parallelism is substantial
- Throughput is more important than latency

Parallelism is an integral part of GPU computing and will become important in future. GPUs offer parallel computing power that usually requires a computer or a supercomputer

to accomplish. Several applications with substantial parallelism increasingly use the massively parallel computing capabilities of GPUs to achieve superior performance [6]. GPUs are leading the way in parallelism; compilers, algorithms, and computational models have made significant advances in
recent years.

APPLICATIONS

Recently, GPUs are considered not only for graphics, but also for speeding-up the execution of general purpose algorithms. GPU computing has become an increasingly important tool to develop efficient applications in several areas such as computer vision, linear algebra, statistical analysis, physics and biological simulation, database analysis, computational finance, computational biology, and electronic designs. The following represent just a few illustrative applications of GPU computing.

- **Biological Simulation:** Molecular modelling and molecular design take advantage of scientific computation capability of GPU. The performance of GPU-based systems supports the development of custom-made protocols for efficient modelling of biomolecular systems and nanostructures.
- **Computational Electromagnetics:** GPUs are used in a wide range of difference methods or finite-difference time-domain (FDTD) algorithms, finite element methods, moment methods, and Monte Carlo methods in science, engineering, computational electromagnetics, computational fluid dynamics, finance, and acoustics. It is possible to achieve excellent performance with both explicit and implicit computational PDE approximations on GPUs [7]. The familiar MATLAB package can be used to take advantage of GPU computing.
- **Power Systems:** In recent years, the computational demands of modelling modern power systems have been steadily increasing. With the advent of the smart grid, the power system has become more complex and requires more computationally intensive means of simulation and analysis [8].

Other areas of applications of GPU computing include medical physics, operations research, financial engineering, economics, crowd dynamics simulations, optimization, high performance computing (HPC), neuroscience, atmospheric modelling,

BENEFTIS AND LIMITATIONS

The key benefit of GPU computing is its massive performance when compared to the CPU. General purpose GPU computing has produced the fastest supercomputers in the world. GPU computing enables applications that we previously thought impossible due of long execution times. GPUs have the following advantages [9]:

- GPUs are powerful accelerators since they have now hundreds of computing cores;
- GPUs are widely available and relatively cheap;
- GPUs require less energy than other computing devices.

However, GPU computing is only a few years old and the challenges it is facing are daunting. The major challenges include memory, arithmetic, and latencies. To be sustainable, GPU computing must address two open issues: how to increase applications mean time between failures and how to minimize unnecessary energy consumption [10]. In some cases, performance cannot scale with the number of cores because an increasingly large portion of time is spent on moving data rather than performing arithmetic operations. GPUs are not designed to excel at multithreaded access to complex data structures such as a hash table [11].

CONCLUSION

We are in the age of parallel-processing and the age of GPU computing. GPU has attracted a lot of attention and become pervasive in today's computing systems due to its highly parallel and efficient architecture. Using the GPU for computing has been an inevitable trend in scientific community. It is evident that GPU computing will be of great importance in the near future. It seems very promising. More information about GPU computing can be found in [11, 12].

REFERENCES

[1] P. Alevras and D. Yurchenko, "GPU computing for accelerating the numerical path Integration approach," *Computers and Structures*, vol. 171, 2016, pp. 46–53.

[2] A. R. Brodtkorbi, T. R. Hageni, and M. L. Satra, "GPU programming strategies and trends in GPU computing," *Journal of Parallel and Distributed Computing*, vol. 73, no. 1, 2012, pp. 4–13.

[3] P. N. Glaskowsky, "NVIDIA's Fermi: The first complete GPU computing architecture," http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA%27s_Fermi-The_First_Complete_GPU_Architecture.pdf

[4] H. Hsieh and C. Chu, "Particle swarm optimization (PSO)-based tool path planning for 5-axis flank milling accelerated by graphics processing unit (GPU)," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 7, 2011, pp. 676-687.

[5] J. D. Owens et al., "GPU Computing," *Proceedings of the IEEE*, vol. 96, no. 5, May 2008, pp. 879-899.

[6] J. Nickolls and W. J. Dally, "The GPU computing era," *IEEE Micro*, March/April 2010, pp. 56-69.

[7] M. Giles et al., "GPU implementation of finite difference solvers," *Proceedings of the Seventh Workshop on High Performance Computational Finance,* 2014, pp.1-8.

[8] D. J. Sooknanan1 and A. Joshi, "GPU computing using CUDA in the deployment of smart grids," *Proceedings of SAI Computing Conference*, London, UK, July 2016, pp.1260-1266.

[9] V. Boyer and D. E. Baz, "Recent advances on GPU computing in operations research," *Proceedings of the IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum*, 2013, pp. 1778-1787.

[10] J. Y. Shi et al., "Sustainable GPU computing at scale," *Proceedings of IEEE International Conference on Computational Science and Engineering*, 2011, pp. 263-272.

[11] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming.* Upper Saddle River, NJ: Addison-Wesley, 2011.

[12] W. W. Hwu, *GPU Computing Gems*. Burlington, MA: Elsevier, 2011.