Journal of Advanced Computing Research Vol. 3, Issue 2 (2018) 9-12

Parallel Processing Problem and Solution - A Case Study on MATLAB Parallel Computing Toolbox GPU Computing

Fatin Fazila Sulong [#1], Mohamed Faidz Mohamed Said [#2]

[#] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA*
70300 Seremban, Negeri Sembilan

[1] fatinfazilasulong@gmail.com

[2] faidzms@ieee.org

***Abstract*—Parallel computing is the use of at least two processors likes cores or computers in grouping to solve a particular issue. This research is done to study the use of graphics processing unit (GPU) in parallel computing in order to solve the parallel problems and discover its solutions. GPU processing have been utilized as a part of the fields of research and science since the GPU was presented. Many previous research papers are done on the use of GPU in parallel computing and how the GPU is implemented. Other than that, based on the previous research papers, this paper will state some of the types of GPUs used in order to solve the parallel processing problem. Most of them utilize GPUs from NVIDIA despite of the fact that it is the first commercially available GPU for a desktop computer. As the level of programming control of GPUs expand, it is used in the arrangement of broadly useful figuring issues that have delivered the usefulness of GPU computing. This paper summarizes the GPU computing based on the fourteen reviewed research papers. Lastly, this paper will demonstrate that the GPU computing has become a reliable method in solving parallel computing problems.**

Keywords: **Parallel, processing, computing, GPU, MATLAB**

I. Introduction

In numerous years, GPU has been generally used to confront the parallel processing issue with GPU computation. GPUs have been developed from being partly programmable in 1999 to a more practical and exceptionally aggressive option architecture to CPU-based parallel computing in 2011 [1]. Since GPU is better in registering offloads, serious parts of the application is passed on to the GPU, while the rest of the codes still keeps running on the central processing unit (CPU). Besides, in hybrid CPUGPU systems, CPUs and GPUs are used in corresponding link to a heterogeneous co-processing computing model. Computationally genuine parts which can be prepared in a monstrous parallel way are quickened by the GPU because of the preferred standpoint from their high figuring achievement while the CPU takes a shot at consecutive calculations. Overall, the application runs faster and the sharing of tasks makes processing computationally rigorous algorithms very efficiently on GPU. The execution benefit of GPU makes this technology particularly fascinating for scientific applications. An extensive number of present procedures have already been implemented on GPU to link its huge computing control and as a result, more and more researchers are exploring the potential of GPU computing in their respective fields [1]. This study includes the definition, the history of GPU and the summarization of fourteen (14) reviewed paper about parallel computing with GPU computing in another section.

II. Definition

This section will provide some of the definitions related to this study.

A. Parallel Processing

Parallel processing is a process of simultaneously breaking up and running program tasks on multiple microprocessors, thus reducing processing time. Parallel handling might be expert through a PC with at least two processors or by a PC organization. Apart from that, parallel processing is also called parallel computing [2].

B. GPU Computing

According to [3], GPU computing is the use of GPU jointly with CPU to quicken profound learning, diagnostic and designing applications.

C. GPU

For GPU, this paper present two definitions

I. The specialized meaning of GPU is a solitary chip processor with incorporated change, lighting, triangle setup or cutting and rendering motors that is fit for preparing at least 10 million polygons for every second [3].

II. GPU is a processor intended to deal with illustrations operations. This incorporates both 2D and 3D counts, through GPUs principally exceed expectations at rendering 3D designs [4].

III. Historical Background of GPU

A. 1970s

In 1975, the first GPU was introduced which is Fujitsu MB14241. This GPU was utilized to quicken the drawing of illustrations for different 1970s arcade diversions [5].

B. 1980s

In the 1980s, three GPU have been released. Firstly, in 1980 NEC μPD7220 was released and it is one of the main uses of an illustration show controller as a Single Large Scale

Integration (LSI) that included circuit chip, empowering the plan of ease, superior video designs cards [5].

In the 1986, NEC μPD7220 became the best known GPU. Other than that, in this year also, Texas Instruments introduced TMS34010, that is the originally microprocessor with on-chip graphics abilities [5].

The third GPU that has been released in 1980s was IBM 8514 which was one of the first video cards for IBM PC compatibles to execute permanent function 2D primitives in electrical hardware [5].

C. 1990s

In 1991, S3 Graphics presented the S386C911, which its planners name after the Porche 911 as ramifications of the exhibit increment it guaranteed [5].

Next, the primary equipment GPU on home computer game backings was presented in 1996 as Nintendo 64. In the year 1997, Mitsubishi discharged the 3DPro/2MP, a completely highlighted GPU finished of change and lighting for workstations and Windows desktop and ATI used it for their FireGL 4000 realistic card [5].

Towards the end of the 1990s which is in the year 1999, NVIDIA displayed the essential monetarily open GPU for a desktop PC that is called GeForce 256. It could process 10 million polygons per second, allowing it to offload a significant amount of graphics processing from the CPU [5].

D. 2000-2010

ATI Radeon 9700 has been released in 2002. This GPU otherwise called R300 which was the world initially coordinate 3D 9.0 quickening agent, pixel and vertex shades could actualize circling and lengthy drifting point math and were rapidly getting to be as adaptable as CPUs [5].

E. 2010-Present

Radeon HD 6000 arrangement cards were released in 2010. This GPU was utilized as a part of cell phones. Ultimately, Pascal is the most up to date era of realistic card by NVIDIA and was released in 2016 [5].

IV. GPU Processing Strategies

In GPU computing, some of the GPU programming strategies should be considered. Here are some of the GPU programming strategies from reviewed research papers. The strategies are:

1. Rules for latency hiding and string execution [6]
2. Memory strategies [6, 7]
3. Further strategies [6]

For the first strategies, the GPU execution demonstrate is based around the idea of propelling a piece on a lattice comprising of squares and each square comprises of an arrangement of strings, and strings inside a similar piece can synchronize and participate utilizing quick shared memory. Next, the enormously threaded engineering of the GPU is utilized to shroud memory latencies and its dormancy is consequently covered up by the GPU through quick exchanging between strings. Fig. 1 displays the CUDA concepts of grid of blocks.

In memory rules systems, there are three memory regions in GPUs recorded in diminishing request by speed which are registers, shared memory, and global memory [6]. Registers are the speediest memory units on a GPU, and every multiprocessor on the GPU has an enormous, yet lacking, enlist document which is isolated among strings exist in multiprocessor. Shared memory is the second quickest memory sort and it can be as quick as registers if recovered properly. Lastly, the slowest kind of memory on the GPU is the global memory, where it is the primary memory of the GPU. To completely involve the memory transport, the GPU likewise utilizes memory parallelism, in which countless memory demands are utilized to possess the transmission capacity. To achieve the top performance using GPU, correct memory allocation and coreless access are essential [7]. GeForce GTX 590 is an example of GPU computer that has primarily two types of memory which are global and shared memory [7].

Last strategy is further guidelines. In this procedure, the CPU and the GPU are distinctive processors that work asynchronously which implies that it would be able to the CPU and the GPU to perform diverse assignments at the same time, that is an essential part of heterogeneous processing [6].

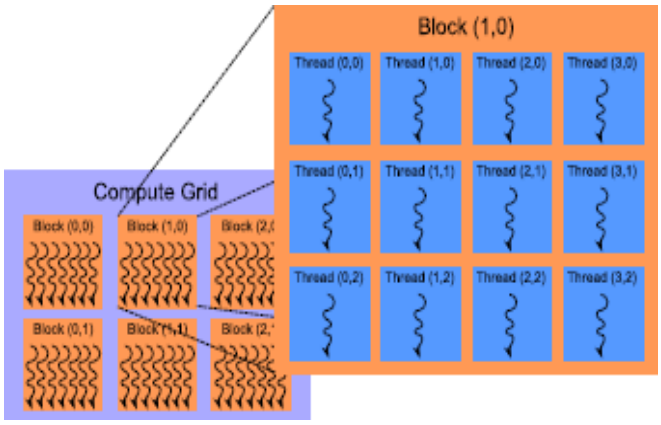

Fig. 1. The CUDA concept of grid of blocks [6]

V. The Use of The GPU

This section will discuss the use of the GPU in order to solve the parallel processing problem and also GPU as a solution in problem solving that includes many areas.

GPU was widely used in order to solve the parallel processing problems. Hence, there are many previous research paper showing about the use of the GPU. Firstly, GPU was utilized as a part of GPU processing to accelerate the coordinated channel techniques, in which the strategies utilize waveforms of distinguished quakes as layouts and look over constant seismic recordings so as to diminish calculations times that were accustomed to recognizing tremors [8]. Other than that [9], GPU is used for constructing the decision tree based on existing parallel computing. Fig. 2 presents the steps in decision tree algorithm.

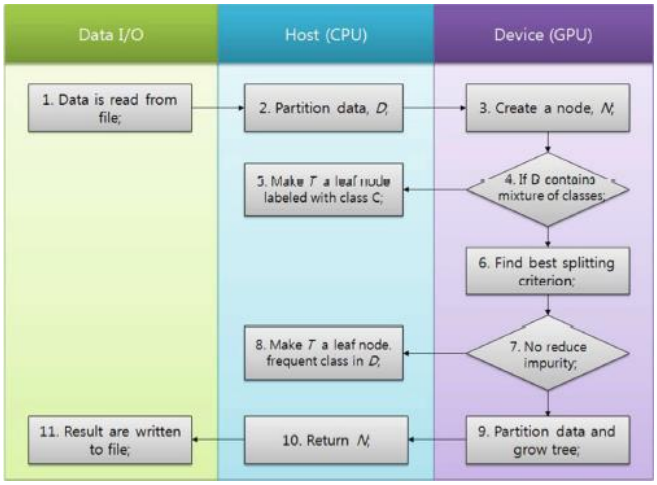

Fig. 2. The step in decision tree algorithm [9]

In addition, a research paper [10] depicts that the GPU is used in parallel algorithm to compute the visibility polygons of an observer point q inside p. In this paper [10], the researchers demonstrate that their calculation can process the permeability of a polygon having more than 4M focuses with the end goal that is displayed in Fig. 3 on a NVIDIA GTX 780 Ti GPU. GPU likewise can be utilized for High Resolution - Finite Volume Method Population Balance Equation (HR-FVM PBE) arrangement utilizing an ease gadget and also to break down the execution of codes with a specific end goal to discover practical exchange off between the exactness and computational expenses, to give a system to the arrangement of single or multidimensional PBEs appropriate for demonstrate based enhancement and ongoing control [11].

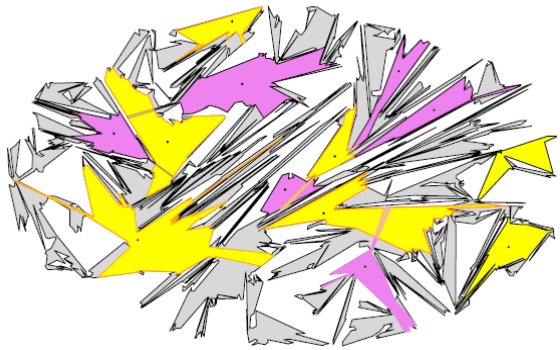

Fig. 3. Visibility polygons of some sample points inside a polygon with 2048 vertices [10]

Besides, paper [12] presents an approach for performing regression on large data sets in reasonable time, using an ensemble of extreme learning machines (ELMs) and implement it on the GPU by NVIDIA. The other use of GPU is to solve large linear systems derived from partial differential equations [13]. Universal Evolutionary Global Optimizer is implemented in CUDA to be run on GPU architectures (GPuEGO) to solve real optimization problems [14].

Moreover, a compelling parallel processing technique that guarantees high optimality and figuring effectiveness in parallel preparing utilizing GPU processor has been proposed in [7]. Lastly, from the reviewed paper, GPU also has been used in determining the possibility of implementing parallel processing in garment drape simulation [15].

VI. Conclusion

GPU can be used to solve the problems of parallel computation. GPU has a reasonable favourable position when contrasted with the CPU. Besides, the computational outcomes exhibit that the GPU execution is five to twenty times speedier than the CPU, contingent upon the specific issue. From the reviewed research papers, it shows that GPUs have been used as a solution to solve parallel processing problems. Other than that, GPU can also be applied in solving the real world problems. Moreover, this study discovers that many researches have been done on the performance of GPU and CPU in solving parallel processing problems in different fields such as scientific and mathematical computing. Some of the research papers also compare the performance of GPU and CPU in order to solve the computational problems. In conclusion, GPU is way better than CPU in order to solve parallel computation problem.

References

[1] G. a. S. F. Wojtkiewicz, "Use of GPU Computing for Uncertainty Quantification in Computational Mechanics: A case study," *Scientific Programming*, vol. 19, pp. 199-212, 2011.

[2] Techopedia. (n.d). *Parallel Processing*. Available: https://www.techopedia.com/definition/4598/parallel-processing

[3] NVIDIA. (n.d). *Accelerated Computing*. Available: http://www.nvidia.com/object/what-is-gpu-computing.html

[4] P. Christensson. (25 November 2016). *GPU Definition*. Available: https://techterms.com

[5] Wikipedia. (n.d 20 May). *Graphics processing unit*. Available: https://en.wikipedia.org/wiki/Graphics_processing_unit

[6] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra, "Graphics processing unit (GPU) programming strategies and trends in GPU computing," *Journal of Parallel and Distributed Computing*, vol. 73, pp. 4-13, 2013.

[7] C. H. Kim and S. Sugano, "A GPU parallel computing method for LPUSS," *Advanced Robotics*, vol. 27, pp. 1199-1207, 2013.

[8] X. Meng, X. Yu, Z. Peng, and B. Hong, "Detecting Earthquakes around Salton Sea Following the 2010 Mw7.2 El Mayor-Cucapah Earthquake Using GPU Parallel Computing," *Procedia Computer Science*, vol. 9, pp. 937-946, 2012.

[9] A. Nasridinov, Y. Lee, and Y.-H. Park, "Decision tree construction on GPU: ubiquitous parallel computing approach," *Computing*, vol. 96, pp. 403-413, 2013.

[10] E. Shoja and M. Ghodsi, "GPU-based parallel algorithm for computing point visibility inside simple polygons," *Computers & Graphics*, vol. 49, pp. 1-9, 2015.

[11] B. Szilágyi and Z. K. Nagy, "Graphical processing unit (GPU) acceleration for numerical solution of population balance models using high resolution finite volume algorithm," *Computers & Chemical Engineering*, vol. 91, pp. 167-181, 2016.

[12] M. Van Heeswijk, Y. Miche, E. Oja, and A. Lendasse, "GPU-accelerated and parallelized ELM ensembles for large-scale regression," *Neurocomputing*, vol. 74, pp. 2430-2437, 2011.

[13] J. M. Elble, N. V. Sahinidis, and P. Vouzis, "GPU computing with Kaczmarz's and other iterative algorithms for linear systems," *Parallel Comput*, vol. 36, pp. 215-231, Jun 01 2010.

[14] J. M. García-Martínez, E. M. Garzón, and P. M. Ortigosa, "A GPU implementation of a hybrid evolutionary algorithm: GPuEGO," *The Journal of Supercomputing*, vol. 70, pp. 684-695, 2014.

[15] I. H. Sul, "Parallel Garment Drape Simulation of Triangular Mesh using

GPU Programming," *International Journal of Clothing Science and Technology*, vol. 21, pp. 252-269, 2009.

[16] F. Sulong. (24 May 2017). *170525 CSC580 FFS*. Available: https://www.youtube.com/watch?v=WnPLiXL_SOI