GPU Computing Revolution: CUDA*

Ramandeep Singh Dehal
Student, DoCST,
MRU Faridabad,
raman19930@gmail.com

Chirag Munjal
Student, DoCST,
MRU Faridabad,
123.munjal@gmail.com

Arquish Ali Ansari
Student, DoCST,
MRU Faridabad,
aliarqish@gmail.com

Anup Singh Kushwaha
Assistant Professor, DoCST,
MRU Faridabad,
anup.usit@gmail.com

***Abstract*—In this research paper, we have studied Compute Unified Device Architecture which was developed by NVIDIA for its GPUs. It is a graphic library that provides a set of APIs' which lets us take advantage of the GPU to render graphics on the computer screen. We have listed the advantages of CUDA over OpenGL & discussed the CUDA process flow & model overview leading to the reviews on its Applications i.e. MRI, Ultrasound, Tomography, X-Rays, CT Scan, DWT-PCA Based Image Fusion, Morphological Image Processing, Parallelizing Multi-Sensor, Pyramidal image blending, parallel wavelet algorithm on GPU using CUDA.**

Keywords—CUDA, GPU, Thread, Block, Nvidia, Kernel, CPU, X-ray, OpenGL, CBCT, MRI, Unified, Radiation, Histogram, Tomography, GeForce, Morphological, Molecular & Vertebra.

I. Introduction

CUDA otherwise called **C**ompute **U**nified **D**evice **A**rchitecture is a parallel computing platform based on a programming model called CUDA C.

It was introduced in 2007 in the 8th series of the Nvidia Graphic Card which had TESLA Micro-architecture. After that, it is also being used in FERMI, KEPLER, MAXWELL & now PASCAL Micro-Architecture Graphic Processing Units (GPUs). Nvidia develops a different API than OpenGL because OpenGL was very focused on the solving specific rendering problems in GPU that include 2D and 3D rendering. Nvidia wanted an API that can focus on other problems like Image Processing Algorithm (such as edge detection, removing clouds, histogram equalization, and Discrete Cosine Transform(DCT) encode and decode etc), Bioinformatics, Molecular Dynamics etc. With the help of parallel computing API it was easy for a graphic card to perform a task efficiently.

It was designed and used by Nvidia Corporations. It empowers an expansion in the processing execution of the Graphic Card by harnessing its energy and is very effective at programming multi-threaded many-cores GPUs. In industry and the scholarly community, it is utilized to accomplish dramatic speedups on creation and research codes. CUDA C is the Application Programming Interface which supports the C and the Fortran programming language. The CUDA C compiler is built on C++.

The advantages of CUDA is that it provides shared memory, it is cost effective & it helps Nvidia to meet the demand of gaming industries in the improvement of GPU.

Although it has disadvantages as well. It is not effective for PC & despite having 100 of "cores" CUDA is not as flexible as CPUs.

OpenGL and CUDA

OpenGL which is a graphic library provides a set of APIs' which lets you take advantage of the GPU to render graphics on the computer screen. OpenGL API contains functions that compile to be run on the GPUs and the code to be run is copied into the GPU and then executed. The program host part does this copying in order to take advantage of the GPU. By the use of standard compilers, the code will be executed only on the CPU. API's present in the library helps you take advantage of this.

Earlier it was thought that GPU was useful only for graphics processing and was used for this purpose only. But in the recent research, it has been discovered that the power of parallel execution of code can also be used for many other problems which include image processing, Bioinformatics, Molecular Dynamics and much more. The main goal of CUDA is to provide the ability for doing general purpose programming on GPU, and its mainly targeted for the NVIDIA family of GPUs. But CUDA doesn't give API's specifically for the rendering of graphics and its processing, so you could re-implement this stuff in CUDA and create your own graphics library with a smarter way and that is OpenGL.

CUDA like, we already mentioned is targeted at NVIDIA graphics cards, while OpenGL will drivers compile the code at runtime and target it for many other graphics cards as well.

Parallel Computing Experience With CUDA

Garland et.al discussed the experience of parallel computing with CUDA. Increase in the parallel process is becoming a primary goal in the CPUs rather than increasing the base clock speed. The old NVIDIA GPUs had multicore processor chips, scaling from 8 to 240 cores. With the arrival of first GPU in view of NVIDIA's Tesla bound unified engineering "NVIDIA GeForce 8800"— it has turned out to be conceivable to program GPU processors specifically and enormously as a parallel processor instead of basically as illustrations API accelerators.

The CUDA parallel programming model comprises of 2 key plan models. In the principal show, it plans to expand a standard consecutive programming dialect, i.e. C/C++ with a base arrangement of reflections for communicating parallelism. In the second model, it is outlined in a route for composing very extensible parallel code that can keep running crosswise over a huge number of simultaneous strings and many processor centers. The CUDA display normally directs the software engineer to compose parallel projects that straightforwardly and productively scale over the diverse

ISBN: 978-1-5386-4119-4/18/$31.00 ©2018 IEEE

levels of parallelism. The parallel programming uses hybrid CUDA, OpenMP and MPI (Message Passing Interface) programming [1].

A CUDA program is executed into a host program, having at least one thread running on the host Central Processing Unit and at least one parallel kernels that are required for the execution on Graphical Processing Unit which is a parallel preparing device. A kernel at that point executes a scalar successive program on a set of parallel threads. The developer needs to arrange these threads into a matrix of thread blocks.

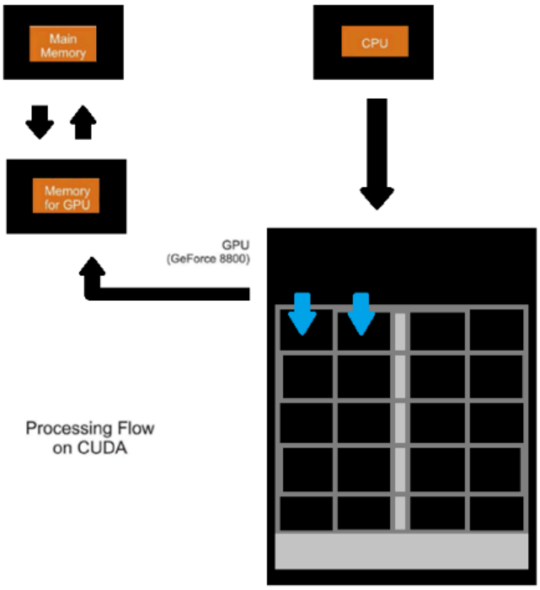

Fig. 1. Process flow in CUDA

The threads of solitary blocks are allowed to synchronize with each other by means of barriers and approach a fast on-chip memory for between threads correspondence. Threads from different blocks in a comparative structure can facilitate just by methods for shared global memory space, which is visible to all of the thread. CUDA requires that thread blocks to be free, inferring that a kernel should execute effectively regardless of the request in which blocks are asked for, regardless of whether all blocks are executed progressively in the arbitrary order. The repression on the conditions between blocks of a kernel gives adaptability. It similarly deduces that the prerequisite for synchronization among threads is the fundamental thought in separating parallel work into isolated kernels. [3]

Process Flow on CUDA

In **Figure 1** we illustrate the process flow in CUDA. The Main Memory copies the data to the Memory of GPU. Then the CPU sends the process instructions to the GPU where it is executed parallelly in each core. After the execution, the data is copied to the Main Memory from the GPU memory.

CUDA model overview

CUDA, **K**ernels, **and T**hreads **:-** The CUDA programming model is a model in which both CPU and GPU are used.
The DEVICE = GPU which execute many threads in parallel.
The HOST = CPU which executes serial portion of an application. KERNELS are the functions that run on the device. The explanation below is illustrated by **Figure 2.**

In CUDA programming model, the GPU is considered as a device. It is a co-processor to CPU and comprise of its own DRAM (GPU memory), which runs numerous threads in parallel. A kernel executes a framework which comprises of blocks. A block comprises of a cluster of threads which cooperate with each other. And the cooperation is done by synchronizing their execution i.e. (shared memory) and by sharing data. Two threads from 2 different blocks cannot cooperate.
Threads and blocks have their own IDs. Using this ID each thread can decide on what data to work on.
This model has solved the memory addressing problems. [2]

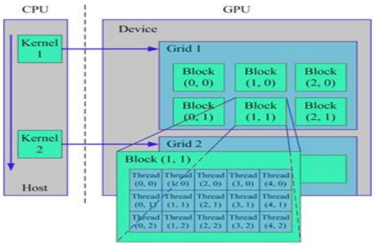

Fig. 2. CUDA Model Overview

II. Applications of CUDA-

a) MRI:

Magnetic **R**esonance **I**maging frames pictures with high spatial-determination, The thing which has to be checked is set in an exceptionally strong field, that makes the turn of its cores adjust either parallel or restricting parallel to the circle. Radiofrequency beats are then connected to energizing the cores, that radiate vitality once they then return to their unique state. The first normal core utilized in imaging is the Hydrogen. To broaden the rate parallel imaging and packed detecting is connected. In X-ray, imaging Gridding is utilized for picture remaking. To dissect the difficulties and chances of using general GPU process, the non-parallel Quick Fourier rebuild algorithmic program was executed by Gregerson that was called as Gridding, on a GeForce 8800 GPU Nvidia's CUDA system. Investigation changes over a manifestation from its unique area (frequently time or space) to the recurrence area and contrariwise. It lessens the nature of processing from O(n^2) to O(n log n). This increase GPU execution by four-hundredth.[4]

b) ULTRASOUND:

Ultrasound guideline is generally utilized for surgical techniques, as a result of its low value and time period visual response. Several ultrasounds target-hunting strategies need giant coaching and simulations over patients. Procedure resources (problems) are typically restricted by time period necessities. In contrast to earlier approaches, they simulate original ultrasound pictures from CT knowledge on the Graphics process Unit. A life scientist “Wein” recommended a technique for calculative properties of tissue and adjust it to a lot of economic forms(computationally). Addition to previous approach, he calculated ultrasound properties from CT knowledge. Exploitation NVIDIA's CUDA a physically adequate reenactment of ultrasound reflection, shadowing (structure like tissue), noticed noise and blurring were made. This will be utilized for mimicking either direct imaging or outspread imaging, and each one determination of the reenacted investigation are intuitively configurable at runtime, as well as ultrasound frequency. With current hardware, we are able to attain a picture breadth of 1023 pixels from raw CT knowledge in the time period, with none reprocessing and loss of knowledge from the CT image. [5]

c) CBCT:

CBCT a therapeutic imaging system comprise of X-beam registered pivotal tomography. CBCT find its application in many field like oral surgery and orthodontia. CBCT is developed inside the clinical space on account of its capacity to create 3D information all through mediations, its high indicative quality, and its short examining circumstances i.e 60sec. Late improvement inside the space of GPUs heps to possess access to unrivaled figuring answer at an espresso value (shabby to utilize),and allowing their utilization in a few logic issues. Noel upheld Relate Nursing recipe for 3D remaking of CBCT by the use of CUDA, that was executed on a NVIDIA GeForce GTX 280. His usage enhanced reproduction times from minutes and hours (efficient) to the seconds. He moreover assessed execution on ten clinical information sets and one apparition learning set to take a glance among processor and GPU. By exploitation, their approach, the calculation time for 2563 is lessened from 25 min to 3.2 s on the GPU. The GPU remaking time for 5123 volumes is 8.5s. [6]

d) X-RAY:

X-radiation (made out of X-beams) is a sort of electromagnetic wave. The X-Ray is a picture inside which the value of each constituent could be a solitary example and assemble the division to be performed. Mahmoudi arranged a division technique which bolstered the Active frame Model. The Active frame Model aides in particular as edges. ASM characterizes an accumulation of structures that portray the frame to be known. This application is postponed because of its calculation time. Mahmoudi explains that the extraction will be performed quickly by using the huge procedure energy of the Graphics procedure Units. Mahmoudi arranged a CUDA-based GPU usage that in result expanded the execution. CUDA uses a bar outline which in turn extends work for X-beam image.To improve the refinement of computerized X-beam CUDA program on a GPU, the bar diagram is frequently acclimated and gets association of the qualification in the image process. To broaden the perceivability of the picture continuously, we tend to utilize the bar graph to extend the work. It's difficult to actualize the task on a GPU because of the propelled technique to exchange the provided information.And in this way the outcome between the memory of GPU is handled. Thus, we tend to show the control on bar outline which extends the work rapidly on GPU by the CUDA program. [7]

e) CT-SCAN:

Computed Tomography (CT) is utilized inside the restorative imaging area, that was started by Wilhelm Konrad Rontgen with the original work on x-beam imaging. It's predominantly a couple of gifts with the end goal that:-

1. It's high spatial determination.
2. Short examining time.

Detriments is the "ionizing radiation", in order to keep up the standard of the picture. The picture denoising strategies are also utilized. Predominantly for two reasons multidimensional CT is extra computationally demanding than 3D CT, for example,

The recreation must be connected to numerous volumes (e.g. 10-20). There are low-measurements strategies because of the amount of radiation is much more than for 3D CT.

As per the Beister et al in 2012, GPU implementation is best suited in CT,The main obstruction is the "radiologists".The work of parallel register provide answer for many issues as it allows the utilization of assets. [8]

f) DWT-PCA Based Image Fusion Using GPU and CUDA:

In their work, Wencheng Wan (2011) considered the CUDA innovation for image fusion as takes after. The source code provided contains both Kernel-GPU and host-CPU code. The host code exchanges information to and from the GPU's worldwide memory and starts the portion (Kernel) code by calling a function. The portion (Kernel) runs various blocks of threads where each thread performs a singular calculation. Threads are sorted out into a chain of importance of matrices of thread block and a network comprises of various blocks that execute a similar kernel. The block comprises of strings that entrance information from the common memory and executes directions in parallel. There is a private neighborhood memory for each thread. Amid the execution, threads may get to information from different memory spaces. [9]

g) Morphological Image Processing on GPU using CUDA:

In their work, D. B. Kirk and W. mei W. Hwu (2010) considered the CUDA innovation for Morphology as takes after. Morphology is clarified as far as set hypothesis. Sets speak to objects in a picture. The Morphological handling is worked by performing tasks on sets of pixels. Normally it is developed for binary pictures. This set involving all the white pixels in a parallel picture is a total morphological depiction of a picture. In the paired (binary) pictures, the sets are given individuals from the 2D whole number space Z2, where every component of a set is a tuple (2D vector) whose directions are the (x, y) directions of a white (or dark) pixel in the picture. The same as above can be stretched out for greyscale pictures. While the Gray-scale pictures can be spoken to as sets, whose parts are in Z3: two segments are directions of a pixel and the third one is its discrete power esteem. A CUDA program is considered as a serial program with parallel segments. This code of CPU and GPU is bound together, it begins its execution on the CPU having serial code and at whatever point a parallel code happens it is executed onto GPU and control is offered back to CPU when parallel code completes execution. The NVIDIA CUDA compiler (NVCC) isolates two unique codes (device and host) and after that incorporates them independently from each other. The host is clarified as a basic C code which is gathered with ordinary C compiler. The code of device is not quite the same as the host code and is likewise customized in a programming dialect with some other little augmentations to have programming dialect. It is assembled with "nvcc". The capacities that execute on GPU and information parallel are called as "kernels". [10]

h) CUDA based parallel wavelet algorithm in medical image fusion:

In their work IMSNA (2013) considered the CUDA innovation for medicinal picture combination (image fusion) utilizing wavelet calculation as takes after. In the present date essentially, CPU and PC group are utilized by the restorative picture combination. At that point the parallel wavelet picture combination calculation is given in view of the model which is CUDA stream figuring model. Furthermore, a while later to the definite examinations of wavelet picture combination, CUDA multi-layer stockpiling structure and SIMD design, the information gathering and the thread task is being deteriorated. The clinic image investigations is additionally done by the assistance of CUDA based wavelet picture combination calculation. In this manner the test demonstrates this calculation can enhance the speed of CPU based one with 8-10 times and it likewise has a linear speed up capacity. [11]

i) Parallelizing Multi-Sensor Image Fusion Using CUDA:

In their work Journal of Computational and Theoretical Nanoscience 7(1):408-411 · March 2012 has considered the CUDA innovation as given takes after. The picture combination (image fusion) in view of Pixel-level would achieve fantastic picture combination and in this way help enhance the execution of picture combination for the multi-sensors. Howsoever there is a vast volume of information that will be prepared yet the continuous combination handling is costly and furthermore un-adaptable. In this way considering the over, a fine-grained pixel-level picture combination execution in view of CUDA (Compute Unified Device Architecture) is proposed which is utilized to quicken Multi-sensor picture combination, in which, every last pixel is alloted to a solitary CUDA thread with the goal that pixel-level picture combination would be parallelized by numerous CUDA threads all the while. The execution for picture combination is additionally enhanced by making utilization of the texture memory in CUDA. What's more, the exploratory outcomes have obviously demonstrated that the proposed strategies would enhance execution amazingly than upgraded CPU partner accessible. [12]

j) Pyramidal image blending using CUDA framework:

In their work, NVIDIA Corporation (2010) considered the CUDA innovation as takes after. The method of pyramidal picture mixing is one of the vital picture mixing system which is utilized as an application in the all panoramic image stitching. The intriguing system of pyramidal picture procedure utilizes various picture pyramids that are required to perform at various resolution levels of pictures. The beginning of the calculation happens with building Laplacian pyramids for the two given (input) pictures and the Gaussian pyramid for the mask picture. A short time later these picture pyramids are joined to frame the Laplacian pyramid, which is subsequently fell to get a seamless panaroma as the yield. The full procedure as specified above isn't just computationally costly tedious also and subsequently it should be speedup, and in this manner therefore, GPU based pyramidal picture mixing calculation is acknowledged on NVIDIA's Compute Unified Device Architecture (CUDA). [13]

III. Conclusion

CUDA is an advance technology that brought revolution in the processing power of the GPUs. The parallel execution of code can also be used to solve many other problems which include image processing, Bioinformatics, Molecular Dynamics.

CUDA C platform is a software layer that has granted direct access to the virtual instruction set of GPUs. CUDA integration is always fantastic as NVIDIA provides high quality support to the app developers who opt for CUDA acceleration. Sometimes CUDA is not as that easy for the various apps to adopt as OpenCL as CUDA is a closed Nvidia framework and not open source like OpenCL but on the other hand, CUDA is supported by a wide variety of apps which is a major advantage of using it. CUDA cross-develops with confidence and ease and thus helps in maintaining and using highly customized environments. It has more mature tools which comprise of a debugger and a profiler which are CUBLAS and CUFFT. Wherever CUDA is integrated it ensures unparalleled performance due to the high-quality NVIDIA support. It provides the user with problem free environment for running jobs.

IV. References

[1] Yang, C. T., Huang, C. L., & Lin, C. F. (2011). Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters. *Computer Physics Communications*, *182*(1), 266-269.

[2] Yang, Z., Zhu, Y., & Pu, Y. (2008, December). Parallel image processing based on CUDA. In *Computer Science and Software Engineering, 2008 International Conference on* (Vol. 3, pp. 198-201). IEEE.

[3] Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., ... & Volkov, V. (2008). Parallel computing experiences with CUDA. *IEEE micro*, *28*(4).

[4] Gregerson, A. (2008). Implementing fast MRI gridding on GPUs via CUDA. *Nvidia Tech. Report on Medical Imaging using CUDA*.

[5] Reichl, T., Passenger, J., Acosta, O., & Salvado, O. (2009, February). Ultrasound goes GPU: real-time simulation using CUDA. In *SPIE Medical Imaging* (pp. 726116-726116). International Society for Optics and Photonics.

[6] Noël, P. B., Walczak, A. M., Xu, J., Corso, J. J., Hoffmann, K. R., & Schafer, S. (2010). GPU-based cone beam computed tomography. *Computer methods and programs in biomedicine*, *98*(3), 271-277.

[7] Mahmoudi, S. A., Lecron, F., Manneback, P., Benjelloun, M., & Mahmoudi, S. (2010, September). GPU-based segmentation of cervical vertebra in X-ray images. In cluster *Computing Workshops and Posters, 2010 IEEE International Conference on* (pp. 1-8). IEEE.

[8] Beister, M., Kolditz, D., & Kalender, W. A. (2012). Iterative reconstruction methods in X-ray CT. *Physica medica*, *28*(2), 94-108.

[9] Jie-Lun Chiang, "Knowledge based Principal Component Analysis for Image Fusion" International Journal of Applied *Mathematics and Information Sciences* 8, No. 1L, 223-230 (2014)

[10] Pierre Sollie, Edmond J. Breen, and Ronald Jones "Recursive implementation of erosions and dilations along discrete lines at arbitrary angles", *Pattern analysis and machine intelligence, IEEE Transactions* on, vol 18, no 5, pp. 562-567, may 1996

[11] Jia Liu, Dustin Feld, Yong Xue, Senior Member, IEEE, Jochen Garcke, and Thomas Soddemann, "Multicore processors and graphics Processing Unit Accelerators forParallel Retrieval of Aerosol Optical Depth From Satellite Data: Implementation, Performance, and Energy Efficiency", *IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing*, No. 5, May 201

[12] Gagandeep Kaur and Sharad P Singh, "Image Decomposition Using Wavelet Transform", *International Journal Of Engineering and Computer Science* ISSN: 2319-7242 Issue 12 Dec 2013 Page No.3477-3480.

[13] Peter J. Burt & Edward H. Adelson, "A Multi-resolution Spline with Application to Image Mosaics", *ACM Transactions on Graphics*, October 1983.

Acknowledgment: We would like to express our sincere gratitude to Accendere Knowledge Management Services Pvt. Ltd., for his valuable comments that led to substantial improvements on an earlier version of this manuscript.