A SURVEY ON GPU SYSTEM CONSIDERING ITS PERFORMANCE ON DIFFERENT APPLICATIONS

Dattatraya Londhe[1], Praveen Barapatre[2], Nisha Gholap[3], Soumitra Das[4]

[1]Department of Computer Engineering, University of Mumbai, Gharda Institute of Technology, Lavel, Maharashtra, India
londhedn@gmail.com
[2]Department of Computer Engineering, University of Pune, SKNSITS, Lonavala, Maharashtra, India
pravinbarapatre@hotmail.com
[3]Department of Computer Engineering, KJ College of Engineering, Pune, Maharashtra, India
golap.nish@gmail.com
[4]Department of Computer Engineering, University of Pune, KJ College of Engineering, Pune, Maharashtra, India
soumitra.das@gmail.com

ABSTRACT

In this paper we study NVIDIA graphics processing unit (GPU) along with its computational power and applications. Although these units are specially designed for graphics application we can employee there computation power for non graphics application too. GPU has high parallel processing power, low cost of computation and less time utilization; it gives good result of performance per energy ratio. This GPU deployment property for excessive computation of similar small set of instruction played a significant role in reducing CPU overhead. GPU has several key advantages over CPU architecture as it provides high parallelism, intensive computation and significantly higher throughput. It consists of thousands of hardware threads that execute programs in a SIMD fashion hence GPU can be an alternate to CPU in high performance environment and in supercomputing environment. The base line is GPU based general purpose computing is a hot topics of research and there is great to explore rather than only graphics processing application.

KEYWORDS

Graphics processing, Hardware threads, Supercomputing, Parallel processing, SIMD

1. INTRODUCTION

Inventions and research in technology has always increased human comfort and reduce human efforts. These implicit aims have always motivated researchers to explore different dimension in technology and science. Recently computer technology plays a great role when it comes to excessive computation to solve a special or particular problem. GPUs have been widely used as components of complex graphics application. Nowadays these graphic processing units are gradually making a way into cluster computing system as the high performance computing units, due to their prominent computational power.

Before when CPU was only the unit for computation many task had to wait for their completion, gradually the idea of processor clustering came into market which not only increased performance but also provide ease for complex computing. Clustering of processor proved to be beneficial for complex computation but along with its benefits there were some unwanted features like high amount of investment, costly for usage when there is less complex computation. GPUs invention proved to be a boon not only for graphics related application but also for other excessive computational SIMD (Single Instruction Multiple Data) tasks. Over few years GPU has evolved from a fixed function special – purpose processor into a full-fledged parallel programmable processor with additional fixed function special –purpose functionality [1].

GPGPU (General Purpose GPU) is study on how to use the GPU for more general application computation and it is gradually increasing [2]. Nvidia announced their CUDA (Compute Unified Device Architecture) system which was specifically designed for GPU programming. It was development platform for developing non graphics application on GPU. CUDA provides a C like syntax for executing on the GPU and compiles offline, getting the favors of many programmers [1]. Nvidia invented GPGPU system known as CUDA in 2006. CUDA allowed programmers to design highly parallel computation program with ease on GPU. CUDA program is mixed code of GPU and CPU. The main routine, complied by the standard C compiler, is generally executed on CPU, while the parallel computing portion is compiled into GPU codes and then transferred to GPU [3]. Functions of CUDA threads is called kernel, n such CUDA threads will perform this kernel n times in parallel.

2. STUDY OF GPU

The first generation NVIDIA unified visual computing architecture in Geforce 8 and 9 series, GPUs was based on a scalable processor array (SPA) framework. The second generation architecture in GeForce GTX 200 GPU is based on a re-engineered, extended SPA architecture [4]. The SPA architecture consists of a number of TPCs which stands for “Texture Processing Clusters” in graphics processing mode and “Thread Processing Clusters” in parallel computational mode. Each TPC is in turn made up of a number of streaming multiprocessors (SMs) and each SM contains eight processor cores also called as streaming processor (SPs) or thread processor [4].Example is NVIDIA G80 GPU, which includes 128 streaming processors. A SM consists of eight streaming processor therefore G80 GPU contains 16 SMs. SM is responsible to carry out creation, management and execution of concurrent threads in hardware with no overhead. This SM support very fine grained parallelism. GPU Parallel computing architecture is featured for parallel computing. The difference between computation mode of CPU and GPU is that GPU is specialized for compute-intensive and highly parallel computation.

For parallel computing the user can define threads which run on GPU in parallel using standard instructions. User can declare the number of threads that can run on a single SM by specifying a block size. User can also state the number of blocks of thread by declaring a grid size, Grid of threads makes up a single kernel of work which can be sent to GPU.

GeForce GTX 200 GPUs include two different architectures -graphics and computing.

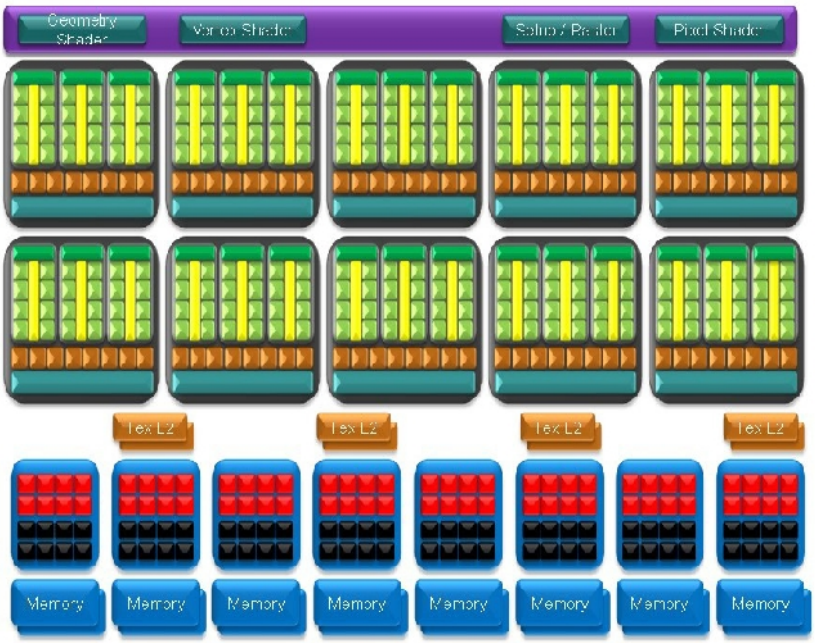

Fig.1: GeForce GTX 280 GPU Graphics Processing Architecture

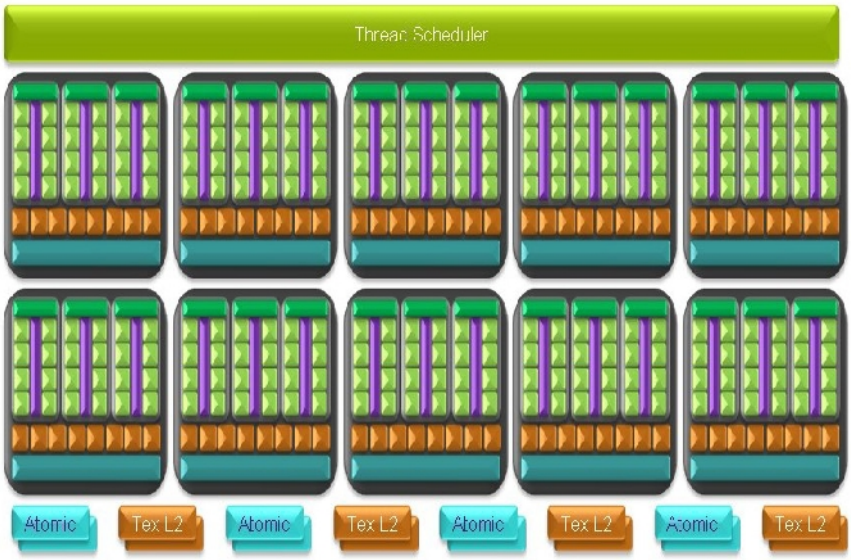

Fig.2: GeForce GTX 280 GPU Parallel Computing Architecture

3. LITTLE ABOUT CUDA

One of the CUDA's characteristics is that it is an extension of C language. CUDA allows the developer to create special C functions, called kernels. Kernel executes on n different CUDA threads. A kernel call is single invocation of the code which runs until completion. GPU follows SIMD / Single Process Multiple Thread (SIMT) model. All the threads are supposed to execute before kernel finishes [5]. CUDA API help user define number of threads and thread blocks.

Each thread block is called CUDA block and run on a single SM. Each thread in a block is synchronized using synchronization barrier. The threads in block are grouped together called a CUDA Warp [5].Memory architecture of CUDA threads is as follows

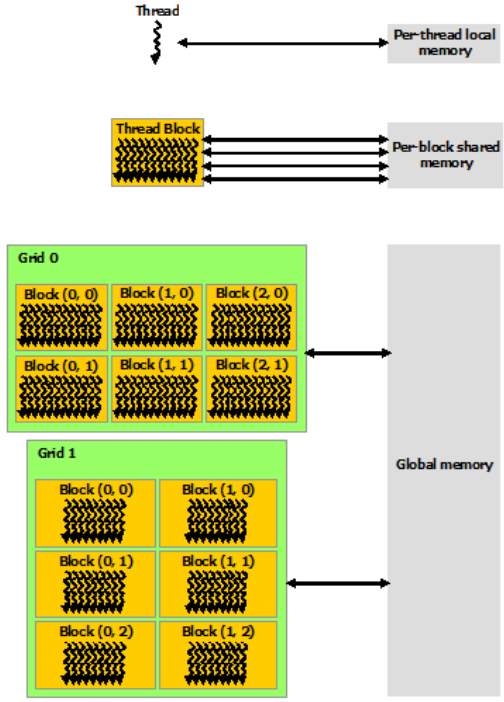

Fig.3. Memory Architecture

Here each thread has private local memory. Each thread block has a shared memory visible to all threads of the block and with the same life time as the block. At last all thread blocks form grids as shown which have access to the same global memory [5].

4. COMPUTATIONAL INTENSIVE APPLICATIONS AND ITS PERFORMANCE ON GPU

4.1 Video Decoding:

When it comes to video or any multimedia application Quality of service become main issue to be handled. Recently people are becoming more and more concerned about the quality of video/visual appliances. GPU units were specifically designed for work such as faster graphics application and better graphics effects, rather than video decoding. In spite of this GPU still proved to be beneficial in partially handling video decoding task. It could be used to perform task that were concerned only with per vertex and per pixel operation. Suppose a block is a regular shape then vertices can be handled by the vertex shader efficiently. Per pixel means all the pixels in a block will go through the same processing. Video decoding highly complex and computationally intensive due to huge amount of video data, complex conversion and filtering process involved in it. The most computational parts in video decoding are Color Space Conversion (CSC), Motion Computation (MC), Inverse DCT, Inverse quantization (IQ) and Variable Length Decoding (VLD). In CSC process every pixel will be translated from YUV space to RGB space using the same equation while for IDCT every pixel will be transformed using different DCT bases as determined by their position [6]. Clearly we can predict that the

most computationally complex MC and CSC are well suitable for the GPU to process both are block-wise and per-pixel operation which IQ, IDCT and VLD are handled by CPU.

CPU and GPU works in a pipelined manner. CPU handles those operational tasks which are sequential, not per pixel type and which may cause more memory traffic between CPU and GPU. So CPU handles operation like VLD, IDCT, and IQ where as GPU handles MC, CSC along with display. This experiment tries to establish CPU and GPU load balance by accommodating a large buffer between CPU and GPU The intermediate buffer effectively absorbed most decoding jitters of both CPU and GPU and contributed significantly to the overall speed-up [6].

We show experimental results of GPU assisted video decoding on pc with an Intel Pentium iii 667-mhz CPU, 256-mb memory and an NVIDIA geforce3 ti200 GPU. This experiment is carried out by uobin Shen, Guang-Ping Gao, Shipeng Li, Heung-Yeung Shum, and Ya-Qin Zhang in paper [6].

Table 1: Experimental Results of GPU Assisted Video Decoding on PC with an Intel Pentium iii 667-Mhz CPU, 256-Mb Memory and an NVIDIA Geforce3 Ti200 GPU

Sequence	Format	Bit rate	Frame rate (CPU only)	Frame rate (CPU + GPU)	Speed-up
Football	SIF (320*240)	2 Mbps	81.0 fps	135.4 fps	1.67
Total	CIF (352 *288)	2 Mbps	84.7 fps	186.7 fps	2.2
Trap	HD 720p (1280 * 720)	5 Mbps	9.9 fps	31.3 fps	3.16

Thus video decoding with generic GPU efficiently increase performance.

4.2 Matrix Multiplication

Some mathematical operations are not practically possible to be solved using pen and paper. The solution for this is use of CPU as a computational device. Mathematical operation like matrix multiplication of huge size matrices lead to overloading of CPU, hence there was degradation of performance. Now the solution is to use either multi-core CPU architecture or GPU. The advantage of GPU over CPU architecture is that GPU is best suited for SIMD operation and matrix multiplication is best example of SIMD. In this application kernel makes up the computation of matrix multiplication on the GPU. Along with the multiplication other initialization are needed to prepare GPU for this computation. These include declaring the thread and block in which the values will be stored [5].

We considered the experiment performed by Fan Wu, Miguel Cabral, Jessica Brazelton in paper [5]. They consider the problem in three stages first is the main file that is recognized by the compiler as a starting point of the program. The second is matrix multiplication algorithm on CPU and the third matrix multiplication algorithm on GPU. After executing the proposed program the result received shows that GPU is much faster than CPU for matrix multiplication. Increase in size of matrix did not give great impact on GPU as that it gave on CPU. Result of this

experiment is shown in the form of graph. This graph represented performance comparison between CPU and GPU based algorithm.

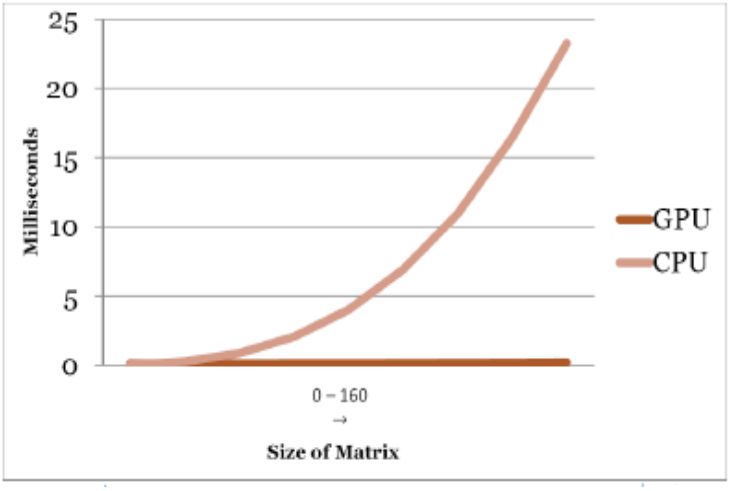

Fig.4: Performance Comparison between CPU and GPU based Algorithms.

4.3 Parallel AES algorithm

Information Security has gains much researcher attention due to increase in threat to important information assets of companies. Data encryption plays important role when it comes to data security. Security is directly proportional to the complexity of encryption algorithm. AES i.e. Rijndael algorithm [7], is a symmetric key cryptography algorithm which is mostly used in data encryption. The traditional CPU-based AES implementation shows poor performance and cannot meet the demands of fast data encryption [8]. AES is block cipher, which divides the plaintext into fixed size blocks. The computation of each block is independent of each other without considering any block cipher mode of operation. When we use CPU for AES encryption each block is encrypted serially. Thus leading to low performance in term of execution time for plaintext encryption, On the other hand GPU executes each plaintext block parallel, thus reducing the encryption time.

We studied Parallel AES algorithm by Deguang Le, Jinyi Chang, Xingdou Gou, Ankang Zhang, Conglan Lu in paper [8]. In this experiment the hardware used are CPU of Intel Core 2 Duo E8200, the memory of 1GB and a GPU graphics card of NVIDIA GeForce GTS250. The software is parallel AES algorithm which runs in windows XP. The result of this experiment is presented in the form of graph which compares speed up of serial AES algorithm and parallel AES algorithm the graph is as follow.

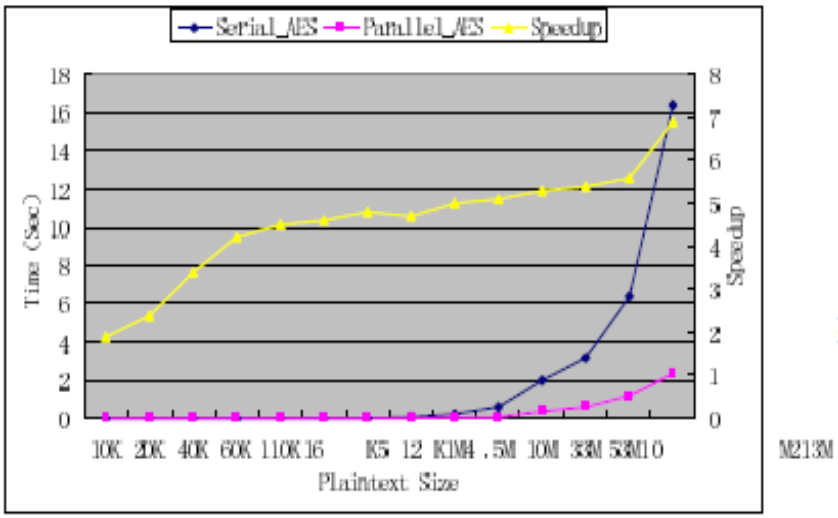

Fig.5: Comparisons of AES algorithms

The speedup is calculated using following formula
Speedup=AES_GPU_Time/AES_CPU_Time

This experiment achieved 7x speedup over the implementation of AES on a comparable CPU [8].

4.4 Password Recovery for MS office 2003 Document

Recently digital data is increasing rapidly. Hence proper organization of data is become an important concern. So at this point MS office comes to picture which helps in properly organizing data files and also providing security by means of encryption and password. MS office 2003 and the previous version organize documents in CFB (Compound File Binary) structure [9]. CFB contain independent data file organize in hierarchy of storage. There are three kinds of encryption scheme available in office 2003 first one is XOR obfuscation, second is 40 bits RC4 encryption and last is CryptoAPI RC4 encryption [11]. Excel puts its encryption information in the 'Workbook Globals Substream'[11]. PPT holds its encryption information with 'CryptSession10Container' in the 'PowerPoint Document' stream [12].

We considered the experiment performed by Xiaojing Zhan, Jingxin Hong in paper [10] states that office document is first analyze then its encryption information are extracted and this is done by CPU after that password verification is to be done, which involve exhaustive calculation having plenty of SIMD task. GPU is involved for password verification. Below the result of experiment is shown in the tabular form.

Table 2: Comparison on time cost between CPU and GPU

Encryption Scheme	Platform	
	CPU(Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz,RAM 2.00GHz,OS-32-bit Windows 7	GPU (GeForce GTX 470)
XOR Obfuscation	<12 min	N/A
40-bit RC4	<64.6h	<4.4h
CryptoAPI RC4	<47.4h	<4.6h

Table 3: Time cost on GPU with different Password Length

Encryption Scheme	Password Length			
	6	7	8	9
40-bit RC4	<4.4h	<11.4d	<705.1d	<119.8y
CryptoAPI RC4	<4.6h	<11.9d	<740.4d	<125.8y

5. GPU LIMITATION

Along with all GPU's advantages there comes some limitation which should be studied while designing any GPU application. Major limitations which can directly or indirectly affect the performance or quality of application are stated as follows.

1. The memory bandwidth between CPU and GPU is limited.
2. Read back from GPU memory to main memory is costly.
3. It has Small instruction set and mainly designed for graphics application.
4. No bitwise operation coding.
5. GPU still missing some features, example is big number support.
6. GPU is not optimized for serial operations.

6. CONCLUSION

From the above study we conclude that GPU is the best alternative for exhaustive computational task. Employing GPU no doubt increase the speed of execution but also frees the CPU from the load to perform serial executable tasks. Combination of CPU and GPU in many applications can render high performance having low cost as compared to cluster of CPUs.

To program GPU the best programming language used is CUDA. It is very efficient and easily understandable programming language to most of the programmer as it is an extension of C language. CUDA programming help in designing heterogeneous computational code which is a combination of serial and parallel execution task performed by CPU and GPU unit respectively.

Many computationally intensive applications have gained benefits from the use of GPU in their computation. There are many more applications under study where researchers are trying to deploy GPU units to gain the best results.

REFERENCES

[1] John D. Owens, Mike Houston, David Lucbke, et al. GPU Computing Proceedings of the IEEE, 2008, 96(5): 879-899.

[2] Zhang Hao, Li Lijun, LiLan General Purpose computation on Graphics Processors [j]. Computer and Digital Engineering, 2005, 33(12):60-62, 98.

[3] Study on GPU based Password Recovery for MS office 2003 Document by Xiaojing Zhan, Jingxin Hong. The 7th International Conference on Computer Science and Education (ICCSE 2012) July 14-17, 2012. 978-1-4673-242-5/12@2012 IEEE

[4] NVIDIA Technical brief. NVIDIA Geforce GTX200 GPU architecture overview second generation unified GPU architecture for Visual Computing. May 2008.

[5] Fan Wu et al., "High Performance Matrix Multiplication on General Purpose Graphics Processing Units" 2010 International Conference on Computational Intelligence and Software Engineering (CiSE), 978-1-4244-5392- 4/10@2010 IEEE

[6] Accelerate Video Decoding With Generic GPU by Guobin Shen, Guang-Ping Gao, Shipeng Li, Heung-Yeung Shum, and Ya-Qin Zhang in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 15, NO. 5, MAY 2005

[7] J.Daemen and V. Rijmen. The Design of Rijndael: AES The Advanced Encryption Standard. New York, USA: Springer-Verlag, 2002.

[8] Deguang Le et al., "Parallel AES Algorithm for Fast Data Encryption on GPU" 2010 2nd International Conference on Computer Engineering and Technology (ICCET), 978-1-4244-6349-7/10@ 2010 IEEE

[9] Compound File Binary File Format [S].Microsoft Corporation, 2010.Available: http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5BMS-CFB%5D.pdf

[10] Xiaojing Zhan and Jingxin Hong "Study on GPU-based Password Recovery for MS Office2003 Document" 7th International Conference on Computer Science & Education (ICCSE 2012) July 14-17, 2012. Melbourne, ustralia

[11] Excel Binary File Format(.xls) Structure Specification[S].Microsoft Corporation,2010. Available: http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/Excel97-2007BinaryFileFormat(xls)Specification.pdf

[12] PowerPoint Binary File Format(.ppt) Structure Specification[S].Microsoft Corporation,2010. Available:http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/PowerPoint97-2007BinaryFileFormat(ppt)Specification.pdf

AUTHORS

Mr. Dattatraya N Londhe received his Bachelor's Degree in Computer Engineering from VPCOE Baramati, Maharashtra, India & is pursuing Master's Degree in Computer Engineering from SKNSITS Lonavala, Pune University, India. He is currently working as an Assistant Professor in Gharda Institute of Technology, Lavel affiliated to Mumbai University. His area of interest is Parallel Computing and information security.

Mr. Praveen R Barapatre received his Bachelor's Degree in Computer Science and Engineering from RGTU, Bhopal & master degree in Remote Sensing and GIS from MANIT, Bhopal. He is currently working as an Assistant Professor & HOD IT in SKNSITS Lonavala affiliated to Pune University, Maharashtra India. His area of interest is Image Processing and Parallel Computing

Miss. Nisha P Gholap received her Bachelor's Degree in Information Technology from Trinity College of Engineering & Research, Pune, & is pursuing Master's Degree in Computer Engineering from KJ College of Engineering & Research, Pune. She is currently working as an Assistant Professor in Gharda Institute of Technology; Lavel affiliated to Mumbai University Her area of interest is Information Security and Parallel Computing.

Mr. Saumitra S Das received Bachelor's Degree in Computer Engineering from North Maharashtra University & Master degree from DY Patil College of Engineering Pune in Computer Engineering. He is currently pursuing Phd in Energy Efficiency in WSN & is working as an Associate Professor in KJ College of Engineering & Research, Pune, affiliated to Pune University, India. His area of interest is WSN and Parallel Computing.