ACCELERATE EXECUTION OF CUDA PROGRAMS FOR NON GPU USERS USING GPU IN THE CLOUD

Prof. Sandip M. Walunj

[1] *Assistant Professor,Computer Engineering Department, SITRC, Nashik,Maharashtra,India*

Abstract – *Existing systems are standalone and can be used by single user at a time. These are inflexible and costly. The main objective of our research is to develop an automated system which will be helpful to the CUDA programmers who doesn't have NVIDIA Graphics card & CUDA installed on their own machines, can take benefit of the parallel processing to enhance the performance of the system by uploading their CUDA programs through their own non GPU based machine to the machine which is having GPU and CUDA installed in the cloud. By using only one high end graphics card and single copy of CUDA installed on the machine in the cloud, non GPU users located far apart from each other can access the GPU for accelerating application execution. It reduces cost and enhance flexibility as compared with multiple users having separate GPU based machine.*

Its costly to buy & install the graphics card on each individual machines separately. Proposed system provides simple UI,higher flexibility, multiuser and high performance within less cost.

Key Words: GPU, CUDA, Computing

1. INTRODUCTION

Now a days, every user of the system always expect fast response from the system. Normally, applications are programmed in serial fashion using programming languages like c, c++, c#, java etc. Serially, developed application takes much more time to complete their execution for massive computations because only one processor is used to perform computation[1]. Because of high performance computing user can get quick response from their system. It becomes possible to the user to do massive computation using parallel computation[2]. There are two types of computations serial and parallel. CUDA programming is the extension of the ANSI C. It provides more number of API to manage dynamic memory in between CPU & GPU. CUDA stands for Compute Unified Device Architecture. It is invented in 2006 to program GPU available on NVIDIA graphics cards. It is used to massive parallel computation using multi-threading. It will help in enhancing the speed of performing computation kind of operations. These computations in details are given below:

1.1 Serial Computing

In serial computing, a program is partitioned into a chiseled order of instructions. Every instruction is executed on a individual CPU one after another. Single instruction executes at a time. Figure-1 represents process of serial computation. Program is a collection of instructions like t0, t1,...,tn. CPU executes t0,t1,...,tn instructions one by one in a sequence. Due to availability of only single processor, same task may not get divided on different processing elements[3]. All these tasks will be executed alternatively one by one. The languages which supports serial computations are C, C++, Java etc. Maximally, serial computing languages supports concurrent execution of tasks on single processor through multi-threading. Figure 1 represents serial computation process in details.

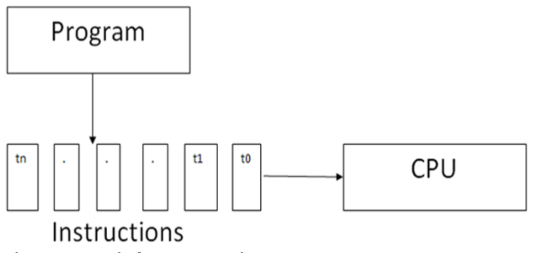

Fig -1: Serial Computing

1.2 Parallel Computing

In parallel computing, multiple processing units are used together to solve a computational problem at a same time. A program is partitioned into number of instructions. Each instruction gets divided into discrete part[3]. Each part of same instruction which is independent on each other gets executed on separate processing units at a time. Instructions are executed using more number of ALU.

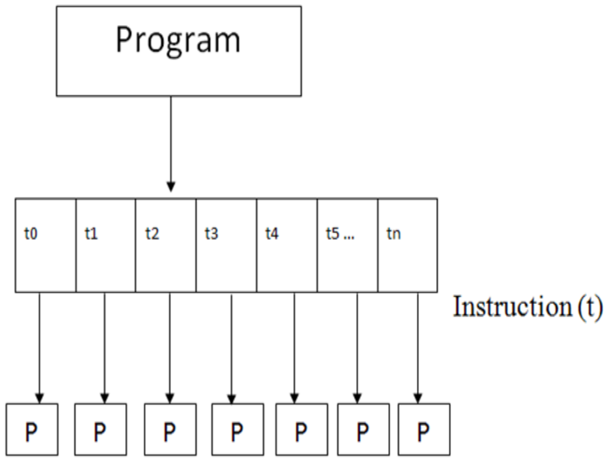

Fig -2: Parallel Computing

2. RELATED WORK

Graphics Processing Units (GPU) have newly increased panoramic popularity among investigators and creators as accelerators for applications outside the domain of conventional computer graphics[4][5]. This development, known as General-Purpose computing on the GPU, vastly results from the improvement in GPU programmability. Everyone is interested to get the fast response from computer for this purpose evolution of High Performance Computing is needed[6]. Immediate response is the need of society whenever large amount of data processing takes place. A Graphics Processing Unit is a many core architecture with each core able to execute cardinals of threads simultaneously. An application with a vast amount of parallelism can use GPU to understand notable performance benefits. GPU is used as a omnipotent surface for performing general purpose high performance computation. Computer programming for GPU is much hard compared to programming general-purpose CPU and parallel programming frameworks such as shared memory programming model.

GPU programming framework renders a general-purpose multithreaded Single Instruction, Multiple Data (SIMD) framework for employing general-purpose computations[7] on GPU. The Compute Unified Device Architecture (CUDA) is such a programming language specifically designed for NVIDIA GPU. It is an extension to C language, CUDA has rapidly become popular and attracted more and more non-graphics programmers to write their applications in CUDA. Still, experience demonstrations that this porting activity becomes difficult task. CUDA places burden on the developer of maintaining GPU and CPU code in isolated functions and need to explicitly handle data movement in between the host memory and GPU memory. It also requires to manually optimize the utilization of the GPU[8][9][10] memory. The investigation involve number of strategies to split computation between GPU threads, of optimizing single-thread code and of appropriately using the GPU memory. It result into significant modifications in the code, the programmer has to do notable changes[11] in the program, may be larger times, before attaining awaited performance. Practically this operation is very hard and error-prone.

Now a days, to enhance the performance of the system by accelerating application execution using parallel processing languages like CUDA, OpenCL[12][13][14] are used on standalone machines.

3. SYSTEM ARCHITECTURE

The main objective of proposing this system is to develop a system which will be helpful to the programmers who doesn't have NVIDIA Graphics card & CUDA installed on their machines can also take benefit of the parallel processing to enhance the performance of the system by uploading their CUDA programs through their own non GPU based machine to the machine which is having GPU and CUDA installed on it situated in the cloud. In this scenario there is need of only one high end graphics card and one copy of CUDA installed on the machine in the cloud and other users who are situated far apart from each other can also access the GPU for speedup their applications. It will cost less amount as compared with multiple users having separate GPU.

To achieve parallel processing using CUDA programming requires NVIDIA Graphics card. The graphics card cost approximately starts from Rs. 2000/- up to lacs of rupees and it depends on the kind of configuration used. To accelerate application performance using NVIDIA GPU on standalone machine it need to have NVIDIA graphics card & CUDA installed on the system. If a batch of 100 students, having 100 different machines. Each student will work on their own machine. To accelerate application running on each students machine it needs to have NVIDIA graphics card & CUDA separately installed on it. It takes more time and cost to buy & install the graphics card on each individual machine separately. Other cost and time efficient solution to this problem is instead of installing CUDA (drivers, SDK & toolkit) and NVIDIA Graphics card on each and every machines separately to enhance the performance, it becomes better to install it on single machine which is available in the cloud and other machines in the network which doesn't have CUDA & graphics card installed on them can also take benefit of it to speedup the application acceleration within the less cost. It provides larger flexibility.

Figure-3 represents proposed system architecture. It consist of three parts - user, non GPU based machines & GPU based machine which is located in the cloud. All machines are interconnected by high speed communication network. The machine in the cloud contains high end NVIDIA Graphics card installed on it. It also contain other software tools required to compile and execute CUDA programs like CUDA(Toolkit, Drivers,SDK). User can login to the system through non GPU based machines and edit their CUDA program using some editor on their local machine. Then user will upload written CUDA program to the machine based in the cloud. Cloud contains a machine having high performance GPU and latest CUDA 6.5 installed on the same for auto compilation and execution of uploaded CUDA programs. The cloud machine compiles the program first, if there are any errors that will be displayed on the users screen. After successful compilation it will execute the program on the GPU so as it will give benefits of parallel processing and helps in improving speed of data computations. After completion of execution of the program it will send result back to the user machine.

Our system will help users who doesn't have NVIDIA graphics card installed on their machine. It saves the cost required to buy the GPU separately and helps in enhancing execution speed of applications. Such services are currently not available in the cloud computing environment. Technologies need to use in the proposed research are cloud computing, parallel processing through CUDA. Currently cloud computing is the most widely used technology for their flexibility, availability, mobility, cost effectiveness features. Parallel processing will properly utilize the potential of available computing resources of computer system like graphics card. Graphics card contains large numbers of processing elements.

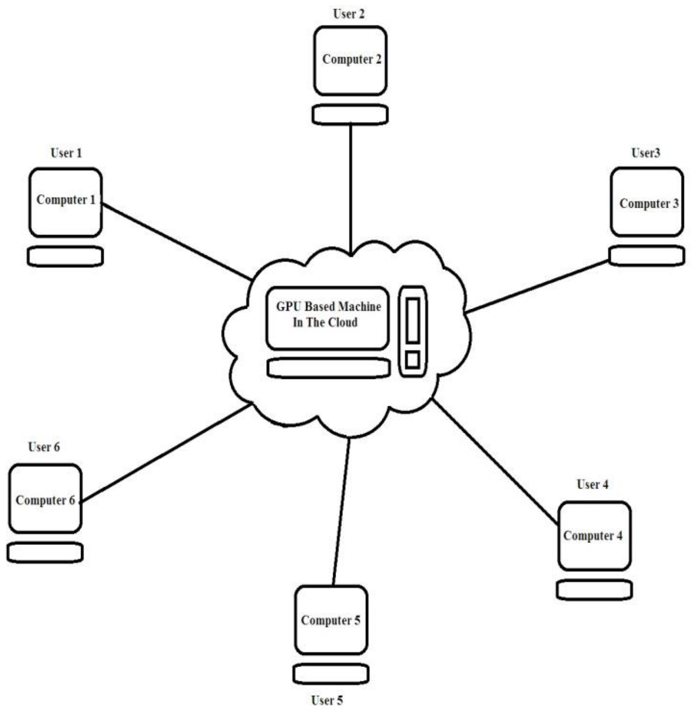

Fig -3: Proposed System Architecture

4. METHODOLOGY

The proposed system is a network of machines where a machine situated in the cloud only can have NVIDIA graphics card & CUDA installed on it. For better performance it requires high speed Internet and high end graphics card. It requires proper scheduling of the job.

The steps to follow:
Step1:User should login to the system.
Step2:Upload CUDA program through local machine to the remote machine in the cloud.
Step3: Compile and execute CUDA program on GPU based machine in the cloud.
Step4: Send result to the client/user machine.

5. CONCLUSION

After completing this research a user who doesn't have NVIDIA GPU installed on their machine can also take benefits to accelerate their applications on cloud GPU. The main outcomes of the system are as follows:

- Users having non GPU based machine can get benefit to accelerate the application execution using GPU installed on machine in the cloud within low cost.
- Saves the cost of buying separate graphics cards to each machine in the network.
- It provides flexibility, mobility because user can access system from any machine, mobile & tablet in the network.
- No need to install CUDA on each and every machine in the network. It will save memory and time required to install CUDA on the machines.

REFERENCES

[1] NVIDIA, "NVIDIA GeForce 8800 GPU Architecture Overview",http://www.nvidia.com/object/IO_37100.html, Nov. 2006.
[2] NVIDIA, "NVIDIA CUDA C Programming Guide v4.2",http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf, April. 2012.
[3] Sandip M.Walunj , Prof. S.A.Kinariwala, "Source to Source Compiler",ELSEVIER, Proceedings of 3rd International Conference on Recent Trends in Engineering & Technology (ICRTET'2014), ISBN No.: 978-93-5107-220-1,Date :28-30 March, 2014.
[4] Taori G, Talole A, Kothawade S, Shrivastava S,"GPU accelerated video conversion in the cloud", IEEE, International Conference on Pervasive Computing(ICPC), 8-10 Jan. 2015.
[5] Rajendra A. Patta, Anuraj R. Kurup, Sandip M. Walunj,"Enhancing Speed of SQL Database Operations using GPU", IEEE, International Conference on Pervasive Computing(ICPC), 8-10 Jan. 2015.
[6] I. Buck et al.,"Brook for GPUs: Stream Computing on Graphics Hardware", Proc. ACM SIGGRAPH, 2004.
[7] S. Ryoo et al., "Optimization Principles and Application Performance Evaluation of a Multithreaded GPU Using CUDA", Proc. Symp. Principles and DISTRIBUTED SYSTEMS, VOL. 22, NO. 1, JANUARY 2011. Practice of Parallel Programming, pp. 73-82, 2008.
[8] C. Liao et al., "Effective Source-to-Source Outlining to Support Whole Program Empirical Optimization," Proc. Int'l Workshop Languages and Compilers for Parallel Computing, Oct. 2009.
[9] S.-Z. Ueng et al., "CUDA-lite: Reducing GPU Programming Complexity," Proc. Int'l Workshop Languages and Compilers for Parallel Computing, pp.1-15, 2008.
[10] J. Fabri, "Automatic Storage Optimization", Proc. Symp. Compiler Construction, pp. 83-91, 1979.
[11] The Portland Group, "CUDA Fortran Programming Guide and Reference",Release 2012.
[12] S. Lee, S.J. Min, and R. Eigenmann, "OpenMP to GPGPU: A Compiler Framework for Automatic Translation and Optimization", Proc. Symp. Principles and Practice of Parallel Programming, 2009.
[13] The Portland Group, "PGI Fortran & C Accelerator Programming Model", Dec 2008.
[14] C.-K. Luk, S. Hong, and H. Kim, "Qilin: Exploiting Parallelism on Heterogeneous Multiprocessors with Adaptive Mapping", Proc. Int'l Symp.Microarchitecture, pp.45-55, 2009.
[15] M.M. Baskaran et al., "A Compiler Framework for Optimization of affine loop Nests For GPGPUs", 2008.
[16] Leonardo Dagum and Ramesh Menon, ―OpenMP: An industry-standard API for shared-memory programming, IEEE Computational Science and Engineering, 5(1):46–55, January–March 1998.
[17] Stone, J.E., Gohara, D., Guochun Shi, ―OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems, Computing in Science and Engineering, Vol. 12, Issue 3, pp. 66-73, May 2010.

[18] E. Alerstam, T. Svensson and S. Andersson-Engels, "Parallel computing with graphics processing units for high speed Monte Carlo simulation of photon migration" , J. Biomedical Optics 13, 060504 (2008).

BIOGRAPHIES

Prof. Sandip M. Walunj,
(ME CSE),
Assistant Professor,
Computer Engg. Department,
Sandip Foundation's
Sandip Institute of Technology & Research Center.
Website:www.sandipwalunj.com