A Learning Approach to Introducing GPU Computing in Undergraduate Engineering Program

Chaker El Amrani
Faculty of Science and Technology, Abdelmalek Essaadi University
Route Ziaten, B.P. 416, Tangier, Morocco

ABSTRACT
The graphics processing unit (GPU) learning initiative is developed within a project awarded by the Moroccan Fulbright Alumni Association (MFAA), entitled "GPU Acceleration of Human Genome Sequencing". This project involves undergraduate students at Abdelmalek Essaadi University, and is conducted in collaboration with the High Performance Computing Lab (HPCL) at the George Washington University in U.S.

The study brings together two of the most important topics and challenges for the medical field, Genomics, and information technology, parallel computing specially with Graphical Processing Units. The potential outcomes from the project will make very valuable contributions to medical and information technology research and will enrich the academic experience of the students.

General Terms
GPU programming, Learning-by-doing.

Keywords
High Performance Computing; GPU; CUDA programming; learning-by-doing; sequences alignment algorithms; bioinformatics.

1. INTRODUCTION
Thanks to a grant from the U.S. Embassy in Rabat, MFAA launched a Call for Proposals for dynamic faculty members in Morocco who are interested in exploring the pedagogical learning-by-doing approach to improve their students' motivation, deep learning and intellectual development.

In such approaches to learning, students solve the problem by seeking the needed knowledge with guidance from their teacher, who plays the role of a coach.

This study stresses the practical use of GPU programming, and the porting of applications to the CUDA programming model particularly sequences alignment algorithms such as Smith-Waterman and Needleman-Wunsch [1, 2].

This project will enable undergraduate students to improve investigation skills, familiarize with the basics of bioinformatics and gain experience in dealing with the rapid changes in computing hardware platforms and languages. In addition, this initiative will help include multicore and GPU programming in the curriculum.

2. THE PROJECT-BASED LEARNING OBJECTIVES
In Genomics, Deoxyribonucleic acid (DNA) sequencing is critical to understanding genetic abnormalities and predicting the risk of developing some diseases, such as Cancers and Alzheimer's [3]. It consists in determining the exact order of the chemical building blocks in a sample. The computation requirement is the highest technical challenge in the Human Genome Project. The Needleman-Wunsch and Smith-Waterman algorithms are used for alignment of DNA Sequences under Global Alignment category. A fast computation solution will be investigated through new-generation graphics hardware.

The advent of low power, massively parallel, programmable NVIDIA CUDA enabled Graphics Processing Units (GPUs) is bringing High Performance Computing (HPC) capabilities to bear on intensive computing genomic applications [4]. As such, the performance improvements provided by GPU computation is expected to dramatically accelerate Needleman-Wunsch and Smith-Waterman algorithms for DNA analyzing, leading to step gains in productivity. Acceleration is made through the use of multi-hundred GPU cores.

Today, HPC technology is ubiquitous. It affects our everyday lives. However, due to limitations in curriculum design, there is still no specific formal teaching HPC at Universities in Morocco.

The project is based on learning-by-doing approach. Students from the Computer Engineering Department will have to attend HPC and GPU seminars, interact with teachers, work in teams and develop CUDA-based software solutions.

The purpose of the project is also to integrate parallel and distributed computing into undergraduate courses.

3. GPU PROGRAMMING
GPUs are offering an outstanding increase in programming performance. GPUs are designed for compute-intensive and time-critical applications, they allow for the execution of threads on a larger number of processing elements. Having a large number of threads may make it possible to surpass the performance of current multicore CPUs [5].

Currently, the environment of computing with GPU is easier for development than few years ago. The two languages CUDA and OpenCL allow programmers to write scripts for GPU without deep knowledge of hardware programming [6, 7, 8].

GPU programming can be used in all areas of graphics and intensive applications that include science, engineering, bioinformatics, art and gaming [9, 10].

CUDA (Compute Unified Device Architecture) is a parallel computing engine in NVIDIA GPUs that is accessible to software developers through variants of industry standard programming languages "Fig. 1". Programmers use 'C for CUDA', compiled through a PathScale Open64 C compiler, to code algorithms for execution on the GPU. Unlike CPUs however, GPUs have a parallel throughput architecture that emphasizes executing many concurrent threads slowly, rather than executing a single thread very quickly.

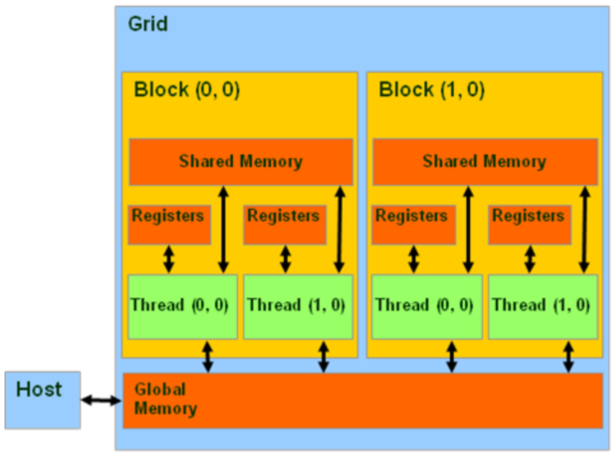

Fig 1: CUDA memory overview

4. DNA SEQUENCING

The DNA, which is composed of four chemical bases, makes all the proteins in an organism.

Virtually every human cell has two strands of DNA, each 3 billion bases long.

When cells replicate, each of the new cells will contain a complete copy of the original DNA. But in case of cancer disease, the results of division are cells that are composed of abnormal DNA or even abnormal numbers of chromosomes.

Alignment of sequences can be made with mathematical algorithms such as Smith-Waterman and Needleman-Wunsch [11, 12]. A speed execution of these algorithms can help cancer prediction and treatment.

We need fast and accurate methods for analyzing DNA of humans and other organisms in order to understand how normal organs develop, and how can cell division lead to birth diseases like cancer, diabetes and Alzheimer.

5. INTRODUCING GPU TECHNOLOGY TO STUDENTS AND ASSIGNMENTS

Seminars were given by the project co-leaders on advances in HPC and programming models. Other presentations detailed CUDA threads and memories, dynamic programming and sequence alignment “Fig. 2”.

According to a survey filled in by the participants, students found the seminars relevant to their studies and research, and simulating their learning. They think that they will be able to use what they learned.

The Project co-leaders discussed the project planning and different phases, they met with students several times, and explained them the project roadmap, objectives and actions to be conducted.

The planning involves permanent meetings, face-to-face and online communications through Forum and mailing-list, frequent student’s feedback, and advisors activities’ follow-up.

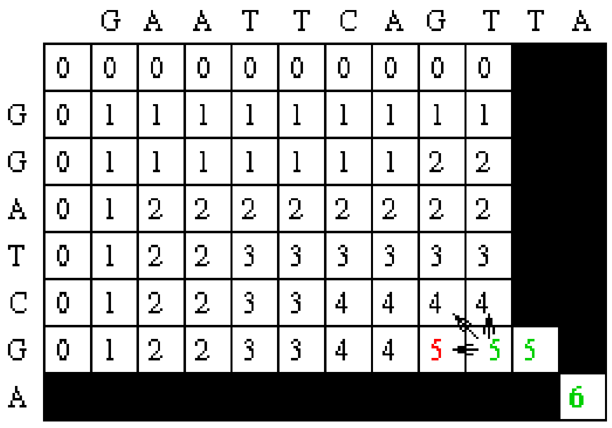

Fig 2: Traceback in dynamic programming

A website was designed to create a social web network for discussion and information sharing [13]. Students can post questions and discuss the assigned tasks. The website’s mean goal is to bring students to a virtual communication framework, and enable them to use mailing-list and Forum, interact with groups, and learn GPU and sequencing techniques. The website provides presentations and courses, project follow up, and information needed to access to the GPU Server.

A GPU Server, HP Z600, equipped with a Quad-Core processor and NVIDIA Quadro FX580 was purchased and configured. Fedora Linux system, NVIDIA driver and CUDA Toolkit package were successfully installed. Students could connect to the Server through Secured Shell protocol. The NVIDIA GPU Computing SDK was implemented. It provides hundreds of code samples, helping students to get started on the path of writing software with CUDA. The NVIDIA card has 32 CUDA cores.

A number of 56 students participated in this project. They had to work in small groups of three to four people, on assigned tasks, and report on progress made. This improves student’s motivation and enhances problem solving and tool use skills in mutli-core and CUDA programming and sequencing technologies [14, 15]. In addition, they develop abilities in both analytical and creative thinking. Students can take these skills with them when they enter the professional world.

6. PORTING ALGORITHMS TO THE CUDA PROGRAMMING MODEL

Each group developed a CUDA program, and compared performance with sequential version, to show up the benefit of using GPU technology. They also prepared and delivered a report describing the investigation methodology, program design and substantial outcomes.

Several groups started with the porting of the following algorithms to GPU, to get practice with CUDA:

- Monte Carlo: a randomized algorithm whose running time is deterministic, but whose output may be incorrect with a certain probability. Monte Carlo methods are often used in simulating physical and mathematical systems.
- Floyd–Warshall: a graph analysis algorithm for finding shortest paths in a weighted graph.
- Histogram calculation: a graphical representation, showing a visual impression of the distribution of

data. Histograms are a commonly used analysis tool in image processing and data mining applications.

- Jacobi: an iterative method of solving a matrix system of the form Ax = b.

For instance, students used Monte Carlo algorithm to calculate the value of PI. They got the result in 20.870 with CPU, and 1.090 seconds with GPU.

Three bioinformatics algorithms were analyzed, designed for GPU, implemented and tested:

- The Needleman–Wunsch algorithm, which is commonly used to align protein or nucleotide sequences. It was the first application of dynamic programming to biological sequence comparison.
- Smith-Waterman algorithm used to compute the edit distance between two sequences typically of either DNA or protein using a dynamic programming approach.
- Levenshtein algorithm that measures the similarity between two strings. There are lots of applications of Levenshtein distance. It is used ,among others, in biology to find similar sequences of nucleic acids in DNA or amino acids in proteins.

7. CONCLUSIONS AND FUTURE PLANS

The Learning-by-doing project succeeded to raise awareness among students and Faculty on parallel computing, and to promote the use of GPU programming and bioinformatics algorithms at the computer engineering Department.

Students gain hands-on experience in multicore programming. They were able to design and implement CUDA programs on GPU Server, and evaluate sequencing algorithms.

GPU programming was consequently included in the undergraduate curriculum this year, through the learning project.

In addition, parallel and distributed computing courses are considered to be included in undergraduate and graduate curriculum with emphasis to multicore and GPU programming.

On the other hand, the implementation of a GPU-based Cluster for teaching and research at Abdelmalek Essaadi University is under study. It will serve for parallel programming hands-on, and for running intensive processing applications.

8. ACKNOWLEDGMENTS

The author would like to thank the Moroccan Fulbright Alumni Association and the U.S. Embassy in Rabat for funding this project. He also wants to extend his thanks to Prof. Tarek El-Ghazawi, Director of the High Performance Computing Lab (HPCL) at the GWU, US, for his great help in supporting this study.

9. REFERENCES

[1] Wen-mei W. Hwu, “GPU Computing Gems Emerald Edition (Applications of GPU Computing Series)”, Morgan Kaufmann Publishers, 2011.

[2] F. Zheng, X. Xu, Y. Yang, S. He, Y. Zhang, “Accelerating biological sequence alignment algorithm on GPU with CUDA”, Proc. International Conference on Computational and Information Sciences (ICCIS 2011), 2011, pp. 18-21.

[3] National Institutes of Health website : http://www.ncbi.nlm.nih.gov/

[4] NVIDIA website: http://www.nvidia.com/object/cuda_home_new.html

[5] D. Kirk and Wen-Mei W. Hwu, “Programming Massively Parallel Processors: A Hands-on Approach”, Morgan Kaufmann Publishers, 2010.

[6] J. Sanders, E. Kandrot, “CUDA by Example: An Introduction to General-Purpose GPU Programming”, Addison-Wesley Professional, 2010.

[7] R. Farber, “CUDA Application Design and Development”, Morgan Kaufmann Publishers, 2011.

[8] C.P. Gribble, “Introducing multithreaded programming: POSIX threads and NVIDIA's Cuda”, Computers in Education Journal 19 (4), pp. 104-112, 2009.

[9] A. Kayi, T. El-Ghazawi, and G. Newby: “Performance Issues in Emerging Homogeneous Multicore Architectures”, Advances in System Performance Modeling, Analysis, and Enhancement. Elsevier Journal: Simulation, Modeling Practice and Theory, Vol 17, Issue 9, pp. 1485-1499, October 2009.

[10] F. Feldhaus, S. Freitag and C. El Amrani, " State-of-the-Art Technologies for Large-Scale Computing", Ch. 1, pp. 1-17, in Werner Dubitzky, Krzysztof Kurowski and Bernhard Schott, Large-Scale Computing Techniques for Complex System Simulations, Wiley-IEEE Computer Society Pr, 2011.

[11] K. Sharma, A. Saxena, P. Kumar, “Alignment of DNA sequence using the features of global and local algorithms along with matrices”, Advanced Materials Research, Volume 403-408, 2012, Pages 2012-2015.

[12] E. Rucci, A.D Giusti, F. Chichizola, M. Naiouf, L.D. Giusti, “DNA sequence alignment: Hybrid parallel programming on a multicore cluster”, Recent Advances in Computers, Communications, Applied Social Science and Mathematics, Proc. of ICANCM'11, ICDCC'11, IC-ASSSE-DC'11 , pp. 183-190.

[13] Project website: http://www.fstt.ac.ma/ginfo/gpu-programming/

[14] D. Díaz, F.J. Esteban, P. Hernández, J.A. Caballero, G. Dorado, S. Gálvez, “Parallelizing and optimizing a bioinformatics pairwise sequence alignment algorithm for many-core architecture”, Parallel Computing, Vol 37, Issue 4-5, pp. 244-259, April 2011.

[15] M. Bailey, S. Cunningham, “A hands-on environment for teaching GPU programming”, SIGCSE 2007: 38th SIGCSE Technical Symposium on Computer Science Education , pp. 254-258.IBM Cloud Computing, Academic Initiative program website: https://www.ibm.com/developerworks/university/cloud/