Exploiting heterogeneous parallel programming for developing an educational neuromorphic computing simulator

Faraz Hussain*, Alvaro Velasquez†, Sumit K. Jha‡
Dept. of Electrical Engineering and Computer Science,
University of Central Florida, Orlando, FL.
*Email: *fhussain@eecs.ucf.edu, †velasquez@eecs.ucf.edu ‡jha@eecs.ucf.edu*

***Abstract*—With the explosive growth of multicore and GPU-based computing, accompanied by the long-predicted demise of Moore's law, there has been an increasing interest in expanding the breadth and scope of undergraduate instruction in concurrent and distributed computing. We describe our plans for exploiting heterogeneous parallel computing architectures for developing an educational simulator for neuromorphic computing. We also report on our experience as early adopters of the NSF/IEEE-TCPP CDER curriculum on Parallel and Distributed computing for Fall 2014 at the University of Central Florida. We include a brief overview of how our courses cover the theoretical and practical aspects of parallel computing, and describe the hardware and software infrastructure used for instruction and research at our institution.**

I. Early adopter experience

The University of Central Florida (UCF) is the second largest university in the country in terms of student enrollment, with more than 59,000 students registering during Spring 2014. In the Electrical Engineering and Computer Science (EECS) department at UCF, we are keenly aware of the stagnation in processor clock speeds that is testing the limits of traditional sequential computing and the accompanying growth in distributed, parallel and multicore computing [1]. We are committed to increasing the exposure of our computer science and engineering students to high-performance computing (HPC) and parallel and distributed computing (PDC) tools. Over the last few semesters, we have been actively incorporating HPC and PDC modules in our undergraduate and graduate algorithms courses, and upper level undergraduate programming and data structures courses. We have also been successfully using big-data problems to motivate students about the importance of parallelism and concurrency, both in theory and practice [2].

Another area that we have focused on is the use of HPC in building, analyzing and validating stochastic computational models [3]. We have used PDC tools and techniques for machine learning statistical models from big data, developing dynamical graphs from structured data, for synthesizing unknown parameters in computational models in control systems and computational systems biology, and to detect communities in evolving complex networks.

For both research and pedagogical purposes, we have made extensive use of parallel programming platforms for simulating models of ordinary differential equations, stochastic differential equations and agent-based models. As a group that specializes in formal methods and model validation and verification, we have used HPC techniques on heterogeneous architectures for giving *statistical* solutions to various *probabilistic model checking* problems.

In this semester's graduate algorithms course, we are emphasizing parallel algorithms, P-completeness and #P-completeness, parallel graph algorithms and neuromorphic computing architectures. The research problems that we are encouraging students to think about include parallel stochastic simulation, simulation of agent-based models over heterogeneous architectures and the use of parallelism for discovery of parameters in probabilistic models.

This summer, NVIDIA selected UCF as a "2014 CUDA Teaching Center" to help promote parallel computing education, and has donated 5 GeForce GTX780s CUDA-capable GPUs and 1 high-end Tesla K40 GPU as well as instructional material in the form of books, slides and lab exercises, access to webinars and promised support for a Teaching Assistant (TA). The EECS department has purchased an educational GPU server that hosts the GPUs given by NVIDIA, and supports two 10-hour TAs with accompanying tuition waivers for core courses that teach parallel and CUDA programming. Our students have also been able to use the Little Fe[1] portable GPU cluster that we received at the HPC educator's workshop co-located with the 2012 Supercomputing Conference.

II. Neuromorphic computing

The construction of a memristor by researchers from HP labs in 2008 [4] has led to an explosion of interest in research in the area [5]. The memristor is a passive, two terminal device that is capable of "remembering" the amount of current that last flowed through it and was hypothesized by Leon Chua in his now widely-known 1971 paper [6]. Over the last five years, there has been a flurry of papers on memristor-based computing in both the popular and scientific press [5]. Memristor have been used for performing arithmetic operations, implementing logical operators for image processing applications [7] and in many other areas [5].

One potential application of memristors deserves special emphasis: the possibility of using memristor based circuits to model the behavior of the human brain [8] due to the

[1]http://littlefe.net/ – "Parallel and Cluster Computing Education On The Move"

property of memristors of behaving like neuronal synapses [9]. Memristor-based technologies have the potential to radically alter the electronics and computing world and cause a seminal shift in the design of computing architecture away from the von Neuman stored program model. The non-volatile storage market is already seeing stamp-sized storage devices that can store up to 1 terabyte of data [10] and complete systems that function with this new paradigm are expected to be available within a decade [11].

Memristor circuits usually come in the form of *crossbars* – two perpendicular arrays of parallel wires with a switch located at each intersection point that acts as a storage system that represents the ON state when closed (by applying a positive voltage) and OFF otherwise [5]. The importance of tools for the simulation and analysis of crossbar designs for memristive systems and networks is likely to grow exponentially over the next decade [12].

We believe that heterogeneous parallel computing platforms should be exploited to build emulators that will be capable of analyzing the behavior of large memristor crossbar designs. We will focus on "neuromorphic computing" [13] – an emerging area that seeks to use memristor-based circuits to model the behavior of the human brain and hence aid in the study of computational neuroscience.

Memristors have demonstrated tremendous potential as the electronic equivalent of the neural synapse [9]. Not only do they possess spike timing dependent plasticity (just like neurons), t their non-volatility provides superior energy efficiency that could bring computing closer to the incredibly low power consumption of the brain. Recent studies have shown that memristor-memory chips can be fabricated with a very high density of 100 GB/cm^2 [14]. Furthermore, the unification of memory and computing units on the same chip mitigates the transfer and latency issues plaguing von Neumann architectures, making memristors ideal as neuromorphic computing elements. These characteristics suggest the memristor-based chip as an excellent candidate for the implementation of a massively parallel neuromorphic computing architecture [15] which we seek to simulate.

The essence of neuromorphic computing is that, in being modeled after the mammalian nervous system, it distributes both storage and computation across a large number of interacting, but independently acting units that communicate through synapses that have an inherent attribute (*synaptic weight*) that determines properties of the communication possible [16], [17]. Memristors, by their very nature, can act as synapses because the amount of charge flowing through them determines their resistance [9]. We are therefore building a robust simulator that can be used to model and analyze memristor crossbars and networks of memristive systems.

III. Summary

Over the next year, we will build a simulator for memristor-based neuromorphic computing meant primarily for instructional purposes. This will aid undergraduate students in their understanding of memristive systems and also help interested research students perform experiments to analyze and validate neuromorphic architectures.

We will continue our emphasis on PDC instruction for undergraduate and graduate students. In Spring 2014, we will teach a graduate level complexity theory course that will cover the PRAM model and complexity of parallel version of many commonly known algorithms. In addition to discussing time-space complexity, decidability, reducibility and intractability this course will also include a module on randomized algorithms that will introduce students to the complexity class ZPP, RP and RNC. We will document all instructional material and course related-information, collect student evaluations for each course, and report to them to the CDER center.

References

[1] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era." *IEEE Computer*, vol. 41, no. 7, pp. 33–38, 2008.

[2] F. Hussain, N. Deo, and S. K. Jha, "Early Adoption – High-Performance Computing for Big Data," in *Procceedings of the Fourth NSF/TCPP Workshop on Parallel and Distributed Computing Education*, Phoeniz, AZ, May 2014.

[3] N. Deo, S. K. Jha, F. Hussain, and M. Vasudevan, "Introducing parallel programming across the undergraduate curriculum through an interdisciplinary course on computational modeling," in *Procceedings of the Third NSF/TCPP Workshop on Parallel and Distributed Computing Education*, Boston, MA, May 2013.

[4] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[5] P. Mazumder, S. M. Kang, and R. Waser, "Memristors: devices, models, and applications," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1911–1919, 2012.

[6] L. O. Chua, "Memristor-the missing circuit element," *Circuit Theory, IEEE Transactions on*, vol. 18, no. 5, pp. 507–519, 1971.

[7] C. K. K. Lim, A. Gelencser, and T. Prodromakis, "Computing image and motion with 3-d memristive grids," in *Memristor Networks*. Springer, 2014, pp. 553–583.

[8] S. Gaba, P. Sheridan, J. Zhou, S. Choi, and W. Lu, "Stochastic memristive devices for computing and neuromorphic applications," *Nanoscale*, vol. 5, no. 13, pp. 5872–5878, 2013.

[9] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[10] Jason Mick, "How Silicon Valley's Best-Kept Secret, Crossbar, Beat HP to the Market w/RRAM," via dailytech.com. Last accessed: 18 September, 2014.

[11] Peter Bright, "HP plans to launch memristor, silicon photonic computer within the decade," Via arstechnica.com. Last acccessed: 21 September, 2014.

[12] H. Kim, M. P. Sah, C. Yang, S. Cho, and L. O. Chua, "Memristor emulator for memristor circuit applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 10, pp. 2422–2431, 2012.

[13] W. Zhao, G. Agnus, V. Derycke, A. Filoramo, J. Bourgoin, and C. Gamrat, "Nanotube devices based crossbar architecture: toward neuromorphic computing," *Nanotechnology*, vol. 21, no. 17, p. 175202, 2010.

[14] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications," in *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*. IEEE, 2009, pp. 485–490.

[15] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study."

[16] D. Monroe, "Neuromorphic computing gets ready for the (really) big time," *Communications of the ACM*, vol. 57, no. 6, pp. 13–15, 2014.

[17] Sean Gallagher, "HP Labs Machine dissolves the difference between disk and memory," via arstechnica.com. Last accessed: 18 September 2014.

Neuromorphic Computing Research Focus The key challenges in neuromorphic research are matching a human's flexibility, and ability to learn from unstructured stimuli with the energy efficiency of the human brain. The computational building blocks within neuromorphic computing systems are logically analogous to neurons. Spiking neural networks (SNNs) are a novel model for arranging those elements to emulate natural neural networks that exist in biological brains. Each â€œneuronâ€ in the SNN can fire independently of the others, and doing so, it sends pulsed signals to other neurons in the netw Heterogeneous computing refers to systems that use more than one kind of processor or cores. These systems gain performance or energy efficiency not just by adding the same type of processors, but by adding dissimilar coprocessors, usually incorporating specialized processing capabilities to handle particular tasks. Usually heterogeneity in the context of computing referred to different instruction-set architectures (ISA), where the main processor has one and other processors have another - usually a Abstractâ€”Neuromorphic computing has come to refer to a variety of brain-inspired computers, devices, and models that contrast the pervasive von Neumann computer architecture. This biologically inspired approach has created highly connected synthetic neurons and synapses that can be used to model neu-roscience theories as well as solve challenging machine learning problems. The promise of the technology is to create a brain-like ability to learn and adapt, but the technical challenges are signicant, starting with an accurate neuroscience model of how the brain works, to nding materials and engi