UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

Electronics and Computer Science

Progress Report: Simulation for Massively Parallel Processors

by

Lingjun Kong

A progress report submitted for continuation towards a PhD

Supervisor: Dr Tom J Kazmierski

Second Supervisor: Professor Steve R Gunn

30 May 2018

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
Electronics and Computer Science

A progress report submitted for continuation towards a PhD

PROGRESS REPORT: SIMULATION FOR MASSIVELY PARALLEL PROCESSORS

by Lingjun Kong

It is a good age for developing a new architecture of computer[1]. Based on the old and developed single-core or multi-core architecture, with the huge computing demand of Artificial Intelligence and increasing computation workload from mobile device and other domains, with the foreseeable end of Moores Law, a more efficient and effective computer architecture is needed. In this report, a review of computer architecture will be included, a thought of how a new architecture based on massively parallel processor could be will be introduced, and a practice for achieving this aim: an incomplete RISC-V project with SystemC will be included.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem and Motivation

Recently years, the Artificial intelligence (AI) has been developed quickly especially since the AlphaGo [2] in 2016. Canonical processor has shown its weakness to produce this kind of massively matrix addition and production operation. GPU, FPGA and many other ASIC AI chip has shown their performance in this domain. It seems like the canonical processor will not able to be developed furthermore. Since 2004, the single core processor met the failed of Dennard-scaling and then turned from single-core to multi-core [3]. The increasing rate of processor performance has slowed down year by year[3]. The power-wall, memory-wall, and even the clock-synchronization has become the bottle-neck of processor performance. Canonical Processor are still occupied the most domain in computing, the slowdown of its performance has limited many domains of society. To combat these problem, we need pay more attention in memory access, energy saving techs and so on.

Massively Parallel Processors (MPPs) is a kind of techs to combine thousands of processors together to produce a huge computation problem. Many-core has shown its performance in energy management[4], such as big.LITTLE architecture of ARM[5]. SpiNNaker Project[6][7] has develop several different scales of event-driven MPPs for brain simulation based on Spiking Neural Network. This kind of architecture could also be used for another domain such as weather simulation and physical simulation. Compare with ASIC chips, this kind of architecture will be more universal. Compare with FPGA, this kind of architecture can be powerful with faster clock speed. Compare with GPU, this kind of architecture can be more energy efficiency. Above all, the aim of MPPs is using simple, cheap and self-organized cores to achieve a desktop-level supercomputer, fill up the black domain between general purpose processor and application specific chips.

1.1.1 The Trend of AI

The algorithm of AI, nowadays the most popular method is deep learning, which combine a massively of matrix computation as a network. Though the work of compression of AI algorithm is still continuing, the mainly computation of AI is still in the scope of matrix processing, or sometimes tensor processing. This kind of computation has taken a higher request for processors. At the beginning of the application of AI algorithm, until 1st generation AlphaGo[2] in 2016, the computation of AI algorithm will always in CPU and other supercomputers. With the successfully of AlphaGo and the development of ImageNet Challenge[8], especially the help of NVIDAs CUDA, most of the work could be down more efficiently in GPU[9][10].Many AI chips were also developed for this computation, such like TPU of Google[11][12] and DaoDianNao chip[13]. FPGA also take a place of AI computation[10][14]. Xilinx and Altera also developed specific library for AI algorithm. Neuromorphic processor such like TrueNorth[15] using analogy circuit and SpiNNaker[7] based on digital circuit with off-the-shelf core. It is hard to say which one is the best, they all have their own advantages and disadvantages with different application and environment. But the challenges are still related with the challenge of the canonical processor: memory, energy performance and price.

1.1.2 The Ceiling of Multi-core

The limitation of Thermal Dissipation(TP) has lead to the dark silicon[16] problem[17]. The cores in a chip could not able work in the same time, and Esmaeilzadeh indicated this might be one reason of the end of multi-core if no driver occur[18]. Whether this statement, TP problem has actually limited the heap performance of one chip and reduce the performance-price ratio. Another problem is latency, this latency mainly appears in processor-memory communication[3]. The gate switching speed is limited by voltage. And the communication speed is limited by light speed. Though the Moores Law is still working, the nanometer techs has actually slow down and wire delay scales is much slower than transistor performance[3]. With the development of nanometer techs, the die size has become smaller year by year. The size of a chip and number of cores will pay less in the price cost of one chip. Dark silicon limits the number working core in one time, but ARMs big.LITTLE architecture limits the working core themselves to saving energy. Eventually, the SpiNNaker chips, using event-driven mechanism, has much larger dark silicon than canonical processors. This shows that, dark silicon is a result of TP limitation, but could also be a method of saving energy. In this aspect, multi-core has much more bright future.

1.1.3 Programming Problem

With the increasing of number of cores, the thread management and task distribution might one complex problem. If all cores produce different thread, or with coarse granularity parallelism, the programming is not a problem. But such like matrix computation, it need fine granularity parallelism and need highly synchronization. Synchronization problem involves the memory, cache coherency, communication and many other question, which makes the system complex. SpiNNaker uses Global Asynchronization Local Synchronization (GALA) and with Spiking Neural Network is an excellent solution as a MPPs. But Not all application is suitable to use Spiking Neural Network, and how to organize the architecture of MPPs will affect the choice of programming model. For some problem, such as memory coherency, can be solved with either hardware or software, how to balance the work of hardware and software is another problem. Parallel programming, or concurrent programming is still a complex work. However, if MPPs could use or transfer math model directly to program, it might be much easy than single core programming.

1.2 Research Challenges

Using MPPs for science simulation, AI algorithm and other kind of application, make a more universal MPPs is a new and unpopular work. For me, my fundamental knowledge is not enough, and has less practice in progress design is the main problem. For this research, firstly, design work is a huge challenge, how to finish the design using more efficient tools, language and with more computer assistant is one problem. Simulation is another problem, MPPs has many cores and the workload of simulation is heavy. How to evaluate the work and how to develop the application is a harder work.

1.3 Research Objective

With this research, one kind of MPPs architecture could be generated hopefully. But more important contribution is the method for design a MPPs. In this progress, inter-chip and intra-chip communication will be researched and a new and effective communication protocol might be designed. Simulation work will be another important contribution. With modern high-level synthesis language such like SystemC, many platforms could be used for the simulation of hardware. An idea of how to balance programming and hardware workload could be down and the conception of programming on MPPs could be more popular.

1.4 Report Structure

For this report, a literature will be put firstly, involves the basic knowledge of processors and parallelism. The simulation and computer-aided design method and tools will be review. Trend of hardware and architecture design will be included. GPU and SpiNNaker will be introduced as case study. Finally, an incomplete SystemC RISC-V model is the result of this period.

References

[1] William J. Dally. Performance analysis of k-ary n-cube interconnection networks. *IEEE transactions on Computers*, 39(6):775–785, 1990.

[2] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[3] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 6 edition, 2017.

[4] Shekhar Borkar. Thousand core chips: a technology perspective. In *Proceedings of the 44th annual Design Automation Conference*, pages 746–749. ACM, 2007.

[5] Edson Luiz Padoin, Laércio Lima Pilla, Márcio Castro, Francieli Z Boito, Philippe Olivier Alexandre Navaux, and Jean-François Méhaut. Performance/energy trade-off in scientific computing: the case of arm big. little and intel sandy bridge. *IET Computers & Digital Techniques*, 9(1):27–35, 2014.

[6] Steve B Furber, David R Lester, Luis A Plana, Jim D Garside, Eustace Painkras, Steve Temple, and Andrew D Brown. Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, 62(12):2454–2467, 2013.

[7] Steve B Furber, Francesco Galluppi, Steve Temple, and Luis A Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

[8] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231, 2012.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[10] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[11] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12. ACM, 2017.

[12] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[13] Yunji Chen, Tao Luo, Shaoli Liu, Shijin Zhang, Liqiang He, Jia Wang, Ling Li, Tianshi Chen, Zhiwei Xu, Ninghui Sun, et al. Dadiannao: A machine-learning supercomputer. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 609–622. IEEE Computer Society, 2014.

[14] Chen Zhang, Peng Li, Guangyu Sun, Yijin Guan, Bingjun Xiao, and Jason Cong. Optimizing fpga-based accelerator design for deep convolutional neural networks. In *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 161–170. ACM, 2015.

[15] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[16] Hadi Esmaeilzadeh, Emily Blem, Renee St Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365–376. IEEE, 2011.

[17] Michael B Taylor. Is dark silicon useful? harnessing the four horsemen of the coming dark silicon apocalypse. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 1131–1136. IEEE, 2012.

[18] Joel Emer, Pritpal Ahuja, Eric Borch, Artur Klauser, Chi-Keung Luk, Srilatha Manne, Shubhendu S Mukherjee, Harish Patil, Steven Wallace, Nathan Binkert, et al. Asim: A performance model framework. *Computer*, 35(2):68–76, 2002.