Chapter 14

Next Generation High Performance Architecture Research Team

14.1 Members

Masaaki Kondo (Team Leader)

Yiyu Tan (Research Scientist)

14.2 Overview of Research Activities

The next generation high performance architecture research team is conducting research and development of a next-generation high-performance computer architecture. Currently, we are mainly focusing on non-von Neumann architectures such as systolic arrays and neuromorphic computers based on the latest advances in device technologies, architectures that can integrate next generation non-volatile memories and/or various types of accelerators into a general-purpose processor, the advancement of scientific simulations by accelerating machine learning computations, and hybrid computing architectures that combine the benefits of quantum computing and classical computing. We are also performing detailed co-design evaluations of the computer architectures noted above as well as the co-design evaluations of algorithms that take advantage of them on the supercomputers K and Fugaku.

Another important aspect of designing future high-performance systems is power consumption. Power consumption is a prerequisite design constraint for developing exascale or next-generation computer systems. In order to maximize effective performance within a given power constraint, we need a new system-design concept in which the system's peak power is allowed to exceed maximum power provisioning using adaptively controlling power knobs incorporated in hardware components so that effective power consumption is maintained below the power constraint. This concept is recently known as hardware overprovisioning. In such systems, it is indispensable to allocate the power budget adaptively among various hardware component such as processors, memories, and interconnects, or among co-scheduled jobs, instead of fully utilizing all available hardware resources. We are researching strategies to improve the power efficiency and total system throughput for future hardware overprovisioned supercomputer systems.

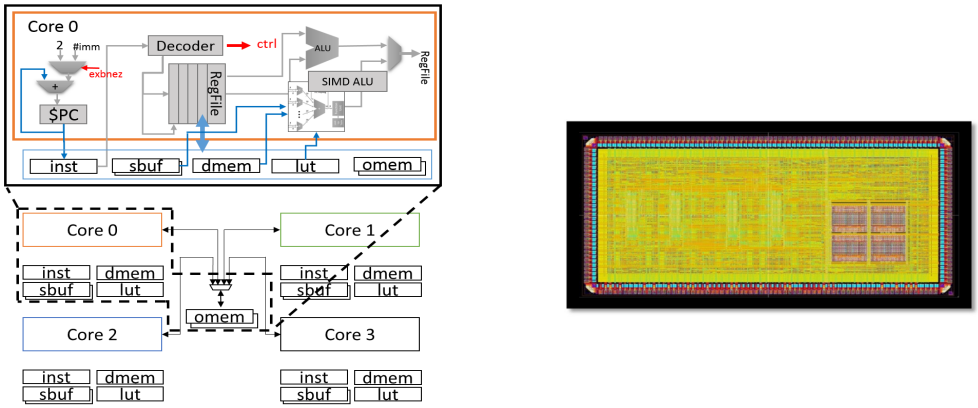
In this fiscal year, we have conducted several researches including evaluations of domain-specific architectures, a prototype implementation of a systolic array in FPGAs, neuromorphic computing for graph processing, and power consumption analysis for low-precision floating-point arithmetic on numerical codes.

14.3 Research Results and Achievements

14.3.1 Real-chip evaluation of a scalable accelerator core for deep neural networks

Recently, a Convolutional Neural Network (CNN) is utilized in many applications such as image recognition and object detection. One of the challenges for executing CNNs is developing a high-performance inference engine with high power efficiency. Several CNN accelerator architectures or LSI chips have been proposed so far for high-performance and low-power CNN executions.

Though existing CNN accelerator architectures can successfully achieve high energy efficiency, they typically focus on optimized execution for either of convolutional or fully connected layers. They also sometimes need to change the network structure which may limit the applicability to the variety of network structures. Since DNN algorithm and organization is now continuing to evolve, it is desirable for CNN accelerators to have flexibility to handle various types of network structure.



(a) Core architecture

(b) Chip layout

Figure 14.1: Core Microarchitecture and Chip Layout of Designed DNN Accelerator

To this end, we have been conducting research and development of the architecture and LSI design of a flexible and scalable DNN accelerator. This is a joint work with several universities in Japan. Our accelerator is a multi-core accelerator with several cores each of which consists of a micro-controller and a SIMD multiply and accumulate (MAC) unit. Figure 14.1 (a) presents the schematic view of the overall accelerator architecture with four cores, each of which has five scratch-pad memories, an instruction memory (inst), a stream buffer (sbuf), a temporal data memory (dmem), a lookup table (lut), and an output memory (omem).

We designed and implemented an accelerator core in real chip. we laid out the chip and taped out it with Renesas Electronics 65nm SOTB process technology. Figure 14.1 (b) shows the layout of the chip. The chip size is 3mm times 6mm with four cores. Due to the chip size limitation, only four cores were implemented on the chip. The chip contains 68-KB of distributed on-chip SRAMs. One core has 2KB instruction and 2KB lookup table memories. The size of dmem and sbuf is 4KB each. The four cores share the 4KB of omem. Each core has 18,8864 logic gates.

The prototype chip was successfully operated and we evaluated performance and power-efficiency of the prototype chip with the LeNet CNN model. We used the MNIST dataset as the image recognition workload. We found that the power consumption is less than 12mW at 50MHz clock frequency. As for performance, we compare the chip with a MIPS R3000 compatible embedded processor which was also developed by our collaborators using the same process technology. We found that our DNN accelerator can achieve 20x higher performance than the general purpose processor.

14.3.2 Neuromorphic graph processing for minimum weight perfect matching

The trend of exponential growth of processor performance known as Moore's law is expected to end in the near future because semiconductor process advancement is almost reaching its physical limit. To achieve further performance improvement in post-Moore era, we need to make use of new types of computer architectures and computing models such as Neuromorphic Computing (NC). The computer systems with NC are attracting a lot of attention as a post-Moore architecture for various reasons. For example, it can potentially mitigate the von Neumann bottleneck and it is inherently power efficient.

In NC, many simple processing elements which are inspired by neurons of a human brain work as computation cores. The communication among them is relatively simple and based on the form of spikes. Therefore, NC has the potential to achieve higher computational efficiency and lower power consumption compared to traditional architectures. Although most of the applications of NC are typically based on neural networks, NC characteristic, massively parallel computation with many simple computational units, can be applied to other types of applications.

We studied on using NC for a graph problem specially a minimum weight matching problem. In particular, we proposed an approximate algorithm for minimum weight perfect matching with NC. We proposed a neuron-like unit named "Spike Delay Neuron" to solve MWPM problems. Figure 14.2 shows the Way to map a target graph to NC with Spike Delay neurons.

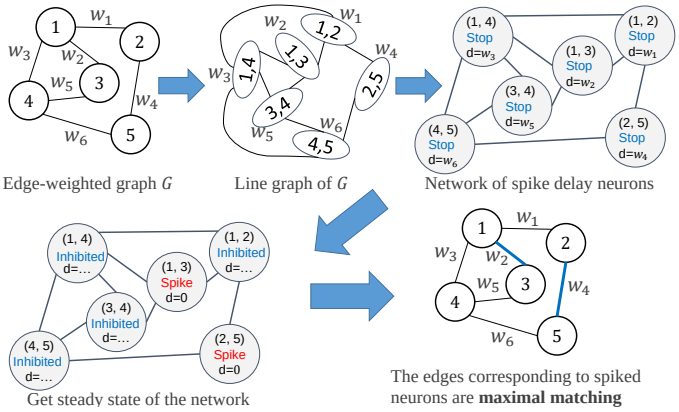


Figure 14.2: A way to solve matching problem by NC with Spike Delay neurons

We also implemented the proposed NC algorithm on an FPGA device. The overview of the hardware architecture is shown in Fig. 14.3. Assuming that the number of neuron units is E, an E-bits register named *spike register* holds spike information of all neuron units. An adjacent matrix and weights of the target graph are saved in a memory which can be either onchip SRAM or off-chip DRAM. The weights are used for the initial value of the spike delay of each neuron.

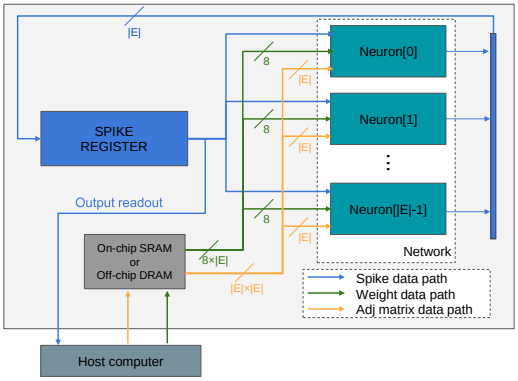


Figure 14.3: Overview of the hardware architecture

We applied it to several random graphs of different scales and concluded that the proposed algorithm is equivalent to a greedy algorithm whose approximation degree is $\frac{1}{2}$.

14.3.3 Power consumption analysis for low-Precision floating-point arithmetic on numerical codes

The low-precision floating point arithmetic that performs computation by reducing numerical accuracy with narrow bit-width is attracting since it can improve the performance of the numerical programs. Small memory footprint, faster computing speed, and energy saving are expected by performing calculation with low precision data. However, there have not been many studies on how low-precision arithmetics affects power and energy consumption of numerical codes. Therefore, we investigated the power efficiency improvement by aggressively using low-precision arithmetics for HPC applications. We analyzed power characteristics of the Poisson's equation and the ground motion simulation programs with double precision and single precision floating point arithmetics.

We changed applications' codes written with double-precision to single-precision and measured execution time, power consumption, and the accuracy of the final result.

In the experiments, we used Reedbush-L supercomputer system installed in the Information Technology Center of the University of Tokyo. Since the effect of using low-precision arithmetics does not change even with parallel computation, we used only a single node. In addition, because power consumption may vary depending on compute nodes to be used due to manufacturing variations, we used a fixed node so that all evaluations are performed on the same compute node. To measure both CPU and DRAM power consumption, we used the Intel RAPL interface which provides power management functionality. It is possible to measure the power and energy consumption of CPU packages and DRAM modules. We create a power measurement library which supports measuring power and energy consumption of a Region of Interest (ROI) portion of the codes. In Poisson's equation solver, a kernel code of the ICCG method was measured. For the earthquake simulation, only the Adaptive CG part is measured. The Reedbush system has two Xeon processors, but we use only one of the processors. The application codes were compiled by Intel compiler with the -O3 option.

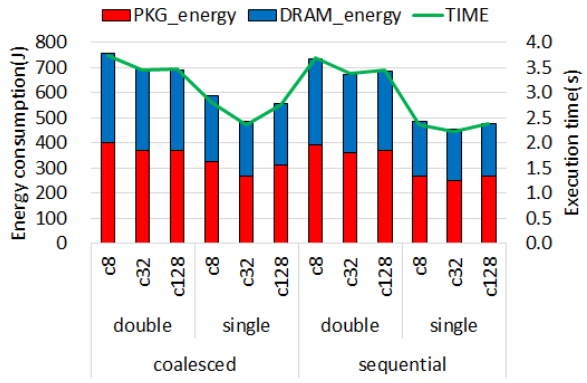


Figure 14.4: Comparison of energy consumption for ICCG solver

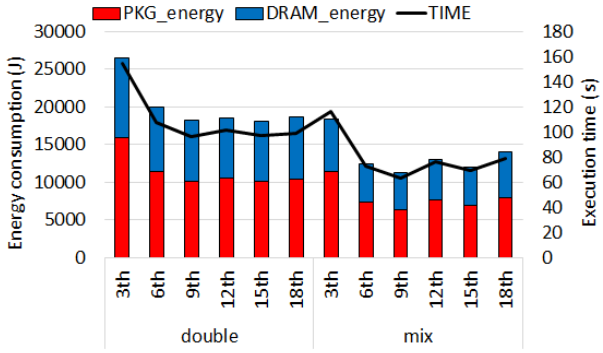


Figure 14.5: Comparison of energy consumption for Adaptive CG

we evaluated the average power, execution time, and the energy consumption comparing the cases of double-precision with single-precision. In the evaluation, for the ICCG solver, we varied the data arrangement method (Coalesced, Sequential) and the number of color divisions (8, 32, and 128 which are denoted as c8, c32, and c128, respectively). For the Adaptive CG code, six cases of the number of OpenMP threads, 3, 6, 9, 12, 15, and 18, were evaluated. Figure 14.4 and 14.5 show the results of the execution time and energy consumption for the ICCG solver and Adaptive CG, respectively.

From the figures, it is confirmed that the execution time is greatly shortened by lowering the compute precision. Since the average power consumption using single-precision did not change very much with the double-precision case, the energy consumption is greatly reduced by using single-precision. As for ICCG, in particular, when the number of colors is 8 (c8) and Sequential is used , energy saving becomes up to 34% in single-precision compared to double-precision. In the case of Coalesced, energy consumption can be reduced by 23.7% on average, whereas in the case of Sequential, energy was reduced by 32.1% on average. If the number of divisions is large, the execution time becomes longer causing the energy efficiency loss.

As for Adaptive CG, the execution time becomes minimum when the number of OpenMP threads is 9 in both cases of double-precision and single-precision even though the evaluated CPU has 18 physical cores per socket. As a result, the best energy efficiency was obtained by using 9 threads with single-precision. Overall, single-precision, we can reduce energy consumption by up to 38.3% and energy saving is 32.4% on average by utilizing single-precision arithmetics.

14.4 Schedule and Future Plan

In order to achieve further performance improvement for next generation HPC systems in post-Moore era, it is necessary to explore various types of devices, hardware architectures, system software/programming models, and algorithms that may contribute to the future system designs. We need to evaluate and analyze huge amount of possible scenarios varying the architectural parameters on wide variety of underlying system architecture. We plan to evaluate several benchmark applications which are expected to become important in future high-

performance computing including big-data and AI as well as traditional simulation applications. We will analyze their performance requirement and execution characteristics. We will also establish a performance model or performance simulation environment that enables to evaluate wide variety of future HPC architectures.

Beside exploring traditional CMOS-based computer systems, we will consider post-CMOS high-performance and low-power computing devices for post-Moore era. We also continue to study an ultra-high-performance accelerator system with an emerging device called SFQ (single-flux-quantum).

14.5 Publications

14.5.1 Articles/Journal

[1] 塚田 峰登, 近藤 正章, 松谷 宏紀, "OSUAD: FPGAを用いたオンライン逐次学習型教師無し異常検知器", 情報処理学会論文誌コンピューティングシステム (ACS65), Vol.12, No.3, pp.34-45, 2019年7月.

14.5.2 Conference Papers

[2] Rei Ito, Mineto Tsukada, Masaaki Kondo Hiroki Matsutani, "An Adaptive Abnormal Behavior Detection using Online Sequential Learning", In the 17th International Conference on Embedded and Ubiquitous Computing (EUC'19), Aug 2019.
[3] Ryohei Tomura, Takuya Kojima, Hideharu Amano, Ryuichi Sakamoto, and Masaki Kondo, "A Real Chip Evaluation of a CNN Accelerator SNACC", In the 22nd Workshop on Synthesis And System Integration of Mixed Information Technologies, Oct. 2019.
[4] Sayaka Terashima, Takuya Kojima, Hayate Okuhara, Kazusa Musha, Hideharu Amano, Ryuichi Sakamoto, Masaaki Kondo and Mitaro Namiki, "A Preliminary Evaluation of Buiding Block Computing Systems", In 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-2019), pp.312-319, Oct. 2019.
[5] Shaswot Shresthamali, Masaaki Kondo, and Hiroshi Nakamura, "Power Management of Wireless Sensor Nodes with Coordinated Distributed Reinforcement Learning", In the 37th IEEE International Conference on Computer Design (ICCD2019), pp.638-647, Nov. 2019.
[6] Ryuichi Sakamoto, Masaaki Kondo, Kohei Fujita, Tsuyoshi Ichimura, and Kengo Nakajima, "The Effectiveness of Low-Precision Floating Arithmetic on Numerical Codes: A Case Study on Power Consumption", In International Conference on High Performance Computing in Asia Pacific Region (HPCAsia 2020), pp.199–206, Jan. 2020.

14.5.3 Posters

[7] Siddhartha Jana, Christopher Cantalupo, Jonathan Eastep, Masaaki Kondo, Matthias Maiterth, Aniruddha Marathe, Tapasya Patki, Barry Rountree, Ryuichi Sakamoto Martin Schulz, Carsten Trinitis, Josef Weidendorfer, "The HPC PowerStack: A Community-wide Collaboration Towards an Energy Efficient Software Stack", In ISC High Performance 2019 poster, June 2019.
[8] Siddhartha Jana, Stephanie Brink, Christopher Cantalupo, Jonathan Eastep, Masaaki Kondo, Matthias Maiterth, Aniruddha Marathe, Tapasya Patki, Barry Rountree, Ryuichi Sakamoto Martin Schulz, Carsten Trinitis, Josef Weidendorfer, "The HPC PowerStack: A Community-wide Collaboration Towards an Energy Efficient Software Stack", In the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'19) poster, Nov. 2019.
[9] Yosuke Ueno and Masaaki Kondo, "Neuromorphic Graph Processing for Minimum Weight Perfect Matching", In the 2nd R-CCS international symposium poster, Feb. 2020.

14.5.4 Invited Talks

[10] Masaaki Kondo, "A Design of Scalable Deep Neural Network Accelerator Cores with 3D Integration", International Forum on MPSoC for Software-defined Hardware (MPSoC'19), July 2019.