Visiting Faculty Program Final Report

Quantitative Evaluation Framework of Machine Learning Processors (MLP) for High Energy Physics

Youngsoo Kim, Assistant Professor, Bradley University

Abstract

As modern high energy physics are producing more data at a faster rate, the need for high-throughput, low latency is becoming primary concern in the development of new read-out architectures. The process of data compression/data selection is applied to across several stages of data pipeline. In recent years, we use machine learning (ML) algorithms increasingly for this use. The goal of this visiting faculty program (VFP) project is to provide an architecture that is specialized for a certain class of ML problems and can therefore deliver a higher level of throughput at a lower power consumption than previously possible. In this project, we selected graph neural networks (GNNs) for acceleration on open-source hardware platforms for optimizing fine-grain irregular data movement ranging from processing elements to the system level. Additionally, we propose a hardware-aware mapping strategy for OpenCL kernels on Vortex GPGPU processors. This method shows optimal hardware resource utilization to achieve better performance and flexibility compared to other mapping approaches. The experiments were validated on 32 different architectural GPU configurations with two typical Graph Convolutional Network layers.

Introduction

As transistor feature sizes have shrunk, physical issues such as crosstalk, wire-resistance, and delay variability are making it increasingly hard to design and optimize a system. To make up for increasing design time, we have no option but to throw more engineers at every project. This problem is evidenced in the “productivity gap” between the number of transistors per-chip that we can effectively manufacture and the transistors-per-designer-per-year that we can effectively design. System-level design techniques promise to improve productivity by designing at higher levels of abstraction. The basic system level design-flow involves describing the application at a high-level of abstraction, exploring the space of possible configurations with a performance-analysis framework, and finally mapping the configuration to an application specific integrated circuit (ASIC) implementation.

The most important aspect of describing a system at an abstract level is that it be fast to simulate. Simulation speed can be increased by separating the details of communication from the details of block description. Within the context of this abstraction, simulation speed can also be increased by abstracting the timing of a system in data flow-oriented language [1]. To date, there has been not much research on this topic in HEP communities and many differing opinions on the best ways to describe, explore, and map systems in this flow. Especially, as new read-out architectures need to process more data at a faster rate, the process of data compression/data selection is becoming primary concern. In recent years, we use machine learning (ML) algorithms increasingly for this use. The goal of most emerging read-out designs is to provide an architecture that is specialized for a certain class of ML problems and can therefore deliver a higher level of throughput at a lower power consumption than previously possible.

The goals for this VFP project are to develop a framework to allow analysis of machine learning (ML) performance at the dataflow-level of abstraction, and to demonstrate this framework on ML hardware accelerators for high level trigger/detector electronics. To this end, we envision a framework that allows AI/ML description at a high-level and mapping to a synthesis flow, allows estimation of throughput, power and area while also permitting the use of very efficient hardware blocks. We envision also that designs must be sometimes carried to lower levels of abstraction in order to get more confidence in the accuracy of their statistics.

Progress

After meeting with Co-PI's Fermilab team including Dr. Farhim and Dr. Guglielmo, we decided to develop a ML framework which maps the graph neural network (GNN) architecture on GPUs. GPUs and NPUs are on the rise of more and more application specific compute architectures, however it requires a close co-optimization between the hardware, the algorithms, the SW-HW mapping for a specific workload, causing several challenges. While large performance improvements are observed in such domain-specific architectures, they often suffer from a lack of versatility across different algorithmic kernels of the future. Moreover, flexible data-parallel compute platforms such as GPUs could be explored or optimized, yet these are typically closed source, hindering deep assessment or further optimizations. Our work demonstrates the impact of HW-aware mapping optimizations of AI workloads on this open-source Vortex GPU. To ensure algorithmic versatility, Graph Neural Network (GNN) are selected as the algorithmic target, as they consist of distinct aggregation and combination kernels.

HARDWARE-AWARE RUNTIME WORKLOAD MAPPING: Vortex POCL compiler accepts standard OpenCL kernels. Compiled code is inserted in boiler-plate assembly, which takes care of initializing the platform and spatially and temporally mapping the parallel instances of the kernel. LLVM takes care to temporally unroll the execution with nested for loops, where the number of iterations of each loop is determined by the local work size (lws), one of the arguments passed by the host platform when calling the kernel execution [3]. Before the execution, the Vortex runtime library maps the compiled kernels across cores, warps, and threads. In the described flow, the spatial unrolling optimizes hardware utilization from cores to threads, in a top-down fashion, taking into account the temporal directives specified by the host. Depending on the relationship between the lws mapping parameter, the algorithmic workload size aws (e.g., the total iterations the kernel will be executed), and the hardware parallelism hp, resolved in Eq. 1, there are 3 possible scenarios: 1) lws < aws / hp: the software will spawn more warps than the hw can support. The execution will be scheduled at different timesteps 2) lws = aws / hp: all warps will be loaded in parallel into the hardware 3) lws > aws / hp: all warps will be loaded in parallel into the hardware, yet with reduced hardware utilization. The optimal lws value is both hardware and algorithm dependent, and can be determined as: lws = aws / hp , with hp = cores × warps × threads. This value can be evaluated at runtime based on the hardware properties and the workload size, without being explicitly specified by the programmer. Figure 1 shows more complex experiments conducted on the vector addition kernel, demonstrating the ability of our mapping strategy to minimize the execution latency. I hand-coded benchmark codes (vecadd, sgemm, gcn, gnn etc) with OpenCL embedding for Vortex. The Vortex is a risc-v based 4x4 GPGPU cores. Vortex POCL compiler take cares of unrolling, where the number of iterations of each loop is determined by local work load sizes. I worked on analytical mapping of this process by hand followed by Vortex simulations for optimal HW-aware mapping. For example, I can validate the mapping of sgemm benchmark for the hardware configuration: 2 cores, 8 warps, 16 threads as in Fig. 1.

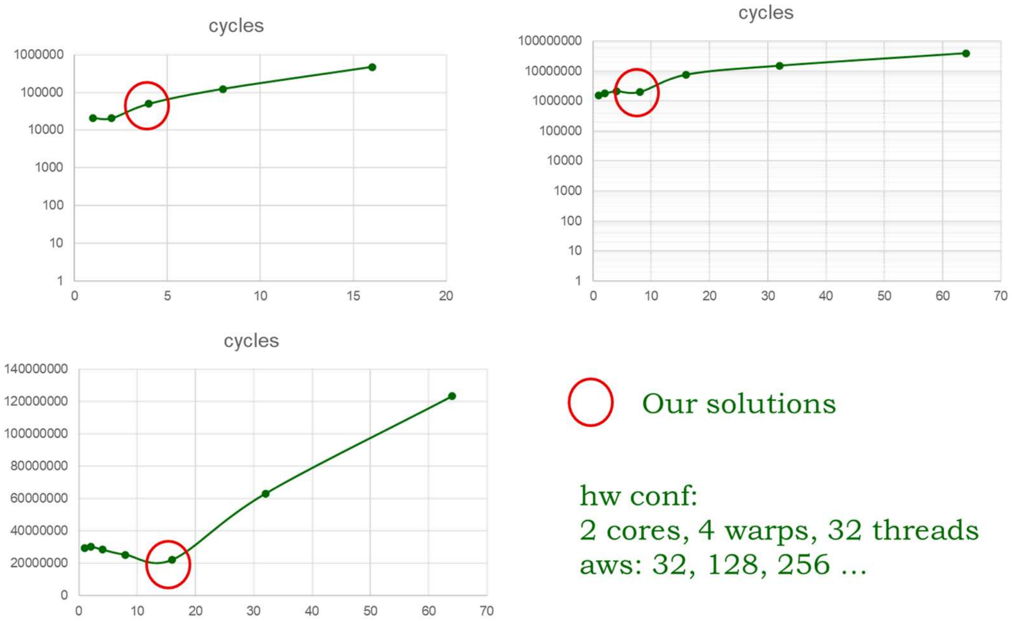

Fig. 1. Example optimal mapping for sgemm

Future Work

Based on the outcome, the scope of this project included GNN kernel mapping results on Vortex gpu, performance metrics with varying gpu configurations, and promising extended instructions for GNN acceleration. Here is the future work planned ahead to extend the work after returning to my home institution.

First, Co-PI, Giuseppe suggested to use ESP platform with risc-v and set up as a loosely coupled accelerator (risc-v + custom ALUs on ESP for an ASIC/RTL implementation). I agreed with his open source approaches. I will study ESP platform and will start to transfer the candidate extensions on ESP. A small custom gpu like processors at L1 seems okay idea but I need to check with Farah and others. Presently, GNNs are accelerated using full size GPUs as more like cloud services. We agreed that GNN acceleration topic at edge device is seemly a good topic and not many work has been done yet. We had open discussions about Giuseppe's active projects including eFPGAs. One of VFP goals is to have me understand what is going on in the lab. So I will keep this relationship going and we can work together for proposal submissions.

Additionally, PI will develop a high performance and radiation hardened configurable fault tolerant processor platforms by mobile customized FPGA platforms. Project activities will provide exciting opportunities for educating undergraduate and graduate students, and particularly underrepresented minority students, in cutting edge hardware and software development.

Impact on Laboratory or National Missions

More specifically, this summer research project attempts to leverage graphic processors parallelism for graph neural network implementations. The proposed quantification mapping platform can be used for various neural network accelerators. Nationally, the research direction aligned with other laboratories such as Warfare Centers - Naval Sea Systems Command (NSWC) Crane Division's mission and can also be used for high radiation environments including the NRL and Department of Energy sites such as Savannah River National Laboratory for remote monitoring.

Conclusions

In this work, we analyzed the resource-aware hardware mapping flow on open source GPU showing a method to runtime optimize the local workloads parameter and abstract its hardware impact to GPU programmer. We validated the approach on the diverse GCN layers, demonstrating the effectiveness of bringing both hardware and workload knowledge into the mapping process of the open source Vortex.

References

[1] J. Duarte et al., "Fast inference of deep neural networks in FPGAs for particle physics", JINST 13 (2018) P07027, doi:10.1088/1748-0221/13/07/P07027, arXiv:1804.06913.

[2] Farah Fahim, Siddhartha Joshi, Seda Ogrenci Memik, Hooman Mohseni: A Low-Power, High-Speed Readout for Pixel Detectors Based on an Arbitration Tree. IEEE Trans. Very Large Scale Integr. Syst. 28(2): 576-584 (2020)

[3] A real-time FPGA-based cluster finding algorithm for LHCb silicon pixel detector, Giovanni Bassi, Luca Giambastiani, Federico Lazzari, Michael J. Morello, Tommaso Pajero and Giovanni Punzi, Published online: 23 August 2021, DOI: 10.1051/epjconf/202125104016

[4] David Xu, A. Baris Özgüler, Giuseppe Di Guglielmo, Nhan Tran, Gabriel N. Perdue, Luca P. Carloni, Farah Fahim:Neural network accelerator for quantum control. CoRR abs/2208.02645 (2022)

[5] iHGNN: Accelerating HGNNs through Parallelism and Data Reusability Exploitation, R Xue, D Han, M Yan, M Zou, X Yang, D Wang, W Li, arXiv preprint arXiv:2307.12765, 2023

[6] Farah Fahim, Benjamin Hawks, Christian Herwig, James Hirschauer, Sergo Jindariani, Nhan Tran, Luca P. Carloni, Giuseppe Di Guglielmo, Philip C. Harris, Jeffrey D. Krupa, Dylan S. Rankin, Manuel Blanco Valentin, Josiah D. Hester, Yingyi Luo, John Mamish, Seda Orgrenci-Memik, Thea Aarrestad, Hamza Javed, Vladimir Loncar, Maurizio Pierini, Adrian Alan Pol, Sioni Summers, Javier M. Duarte, Scott Hauck, Shih-Chieh Hsu, Jennifer Ngadiuba, Mia Liu, Duc Hoang, Edward Kreinar, Zhenbin Wu:

[11] Giuseppe Di Guglielmo, Farah Fahim, Christian Herwig, Manuel Blanco Valentin, Javier M. Duarte, Cristian Gingu, Philip C. Harris, James Hirschauer, Martin Kwok, Vladimir Loncar,

Yingyi Luo, Llovizna Miranda, Jennifer Ngadiuba, Daniel Noonan, Seda Ogrenci Memik, Maurizio Pierini, Sioni Summers, Nhan Tran: A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC. CoRR abs/2105.01683 (2021)
[12] D. Kadetotad, V. Berisha, C. Chakrabarti and J. -S. Seo, "A 8.93-TOPS/W LSTM Recurrent Neural Network Accelerator Featuring Hierarchical Coarse-Grain Sparsity With All Parameters Stored On-Chip," ESSCIRC 2019 - IEEE 45th European Solid State Circuits Conference (ESSCIRC), 2019, pp. 119-122, doi: 10.1109/ESSCIRC.2019.8902809.

[13] Herwig, Christian. Design of a reconfigurable autoencoder algorithm for detector front-end ASICs. No. FERMILAB-SLIDES-20-121-E. Fermi National Accelerator Lab.(FNAL), Batavia, IL (United States), 2020.

[14] Catapult High-Level Synthesis, https://eda.sw.siemens.com/en-US/ic/ic-design/high-level-synthesis-and-verification-platform/

Appendix.

A. Participants

Name	Institution	role
Dr. Youngsoo Kim	Bradley University	PI, ML processor for HEP design
Dr. Farah Fahim	Fermi lab	Co-PI, HEP algorithm design
Dr. Giuseppe Di Guglielmo	Fermi lab	HEP algorithm and arch. co-design, GNN detailed design
Davide Braga	Fermi lab	Advising on HEP ASIC design, ASIC timing parameters derivations

Scientific facilities

The basic reconfigurable computing system is configured with Titanium Intel Tower PC Servers, each containing two 6-core Intel Xeon E5-2620 processors. Consequentially, there will be a total of 240 processor cores available to project participants. The field programmable gate array (FPGA) boards contain configurable processing elements that are loaded with hardware specifically designed for each application. Each FPGA has a maximum logic capacity significantly larger than 10 million logic gate equivalents.

Notable Outcome

PI expects to extend this work and submit the extended manuscript in open source hardware workshop at the International Symposium on Computer Architecture (ISCA) 2024. Additionally, he will work with NPS Space System Group and Dr. James H. Newman to develop a co-proposal with Fermi lab to lead in in prototyping, manufacturing, and modification of fault tolerant processors.

Research Vibrancy

This hardware mapping framework can be extended and the GNN accelerator can be designed on ESP platform with collaboration with Fermi lab ASIC division. This can be a good end-to-end execution of GNNs on a hardware-software combined architecture. PI will validate the approach on diverse GCN/GNN layers and demonstrate the effectiveness of the framework. PI will perform research engagement with Fermilab ASIC division. Currently, he is a recipient of DOE MSIPP funded AI Internet-of-things project, and he plans to develop a proposal out of this VFP project and submit it to Oak Ridge National Lab in relevant funding solicitation.

Connection to Programs at Home Academic Institution

The educational activities derived from this visiting faculty-ship will make aspects of completed computer architecture research more accessible to students; promote research initiatives among undergraduate and graduate students, and leverage interdisciplinary collaborations between Bradley University and Fermi lab. The education/outreach agenda is designed to foster an environment for developing a new generation of electrical engineers who will take a keen interest in hands-on hardware/ASIC design. The PI will provide research opportunities for undergraduate and graduate students at Bradley University.

Specifically, PI will enrich undergraduate and graduate courses related to AI/ML computing and algorithms. He already updated the curriculum to enhance an introductory AI undergraduate course in microprocessor-based systems and embedded systems. He plans to update two graduate courses in AI/ML computing and AI architecture systems with a hands-on laboratory. More advanced courses have updated contents, emphasizing discussion of recent papers and graduate-level projects.

Designing a hands-on laboratory: A critical component of AI/ML embedded system education program is to supplement lectures with hands-on laboratory sessions. A cloud-based virtual laboratory will provide a high degree of accessibility, flexibility, reusability, and scalability. Using graphical topology and statistical information about the embedded system, students will be aware of actual embedded system conditions when they do a lab exercise. We will provide students with an in-depth understanding of security problems through our labs through their experimentation.