Hardware Architectures for processing Artificial Intelligence and Machine Learning processes

Prof. Asadinia

By
Isaiah Martinez
Summer Shin

December 16 2024

1 Traditional Architecture Artificial Intelligence Chip Technology

This publication is authored by Qinze Jiang and Jiajun Zhan who presented their work at the 2nd International Conference on Computer Science and Management Technology (ICCSMT) held in 2021.

The main goal is this paper is to try to compare different hardware architectures for handling AI and ML tasks and looking at various ways to improve performance for each. There are a total of 3 different types of architectures discussed:

- GPUs
- FPGAs
- ASIC Chips

These different technologies all have different purposes for being used. GPUs are the classical solution to handling ML tasks. They are capable of handling the complex tensor computations due to their many simplistic designed cells (processing units). This simplistic design is contrasted by that of the FPGA, which is more complex and allows them to be programmed via logical blocks (gates) and can be configured before usage. This enables them to be more efficient than the classical GPU which is more general purpose. An alternative to both would be the ASIC chips which are specifically designed to the purpose instead of being programmed for it. By instead designing the hardware for the particular task for AI/ML processing, this can further improve performance.

To improve each of these technologies, the authors propose a series of solutions. For GPUs, they suggest combining their operation via more sophisticated software such as Kubernetes and Docker. For FPGAs, they suggest to use FPGA Accelerators to improve performance for a particular type of ML model. For ASIC chips, they suggest a design using Word Lines and particular charges on each side of the line for maintaining a highly optimized form of calculating Matrix vector multiplication.

The main weaknesses of this paper are that it contains some false information, as Multiple GPUs don't need to be connected via Docker/Kubernetes, and that it focuses primarily on only Neural Networks. Perhaps other ML algorithms may see some improvement so these technologies would be used more regularly. This paper does excel at providing several viable implementations of these various technologies to improve performance.

2 NVIDIA Hopper H100 GPU: Scaling Performance

This paper is presented by Jack Choquette from NVIDIA and is published by the IEEE Computer Society in 2023.

This paper focuses specifically on GPUs and how they can be improved to be utilized for ML tasks. The particular model that is discussed is the novel H100 generation of GPU hardware architecture. This applies improvements by featuring larger L2 caches, more HBM3 sites, higher memory bandwidth, and twice of the throughput when compared to the previous generation A100 GPU.

The author suggests that the new H100 architecture, named the Hopper architecture, is capable of handling better data and parallel execution. By improving the spatial locality of the logical sections on the chip, i.e. the placement and connections between the functional components, they can vastly improve it's performance. This is further improved when connected to another novel technology created by NVIDIA, the NVLink network interconnect. The NVLink is a network of interconnect switches alongisde the hardware CPU and GPU hardware to improve the way that the hardware utilizes the data on the network.

In the H100 architecture, they add a fourth layer to the structure of how it operates to oversee the thread block clusters. This new layer utilizes the resdesign changes of the chipset, and allows for a more sophisticated connection to the network via NVLink. This allows for faster asynchronous data transfer and execution, which allows for major improvements with DL models and other tensor-based calculations.

This paper relies heavily on NVIDIA's previous technology as a basepoint to then extend from. As a singular product from NVIDIA, this is massively beneficial, but seeing as this is proprietary information only held and utilized for NVIDIA means that other hardware developers cannot utilize these improvements directly to their own devices. As such, it remains subject to scrutiny as to whether the presented metrics are truthful.

To give NVIDIA fair credit, they have outlined a novel model for designing GPU chips. These chips can directly connect to the network through a series of interconnected network switches, most likely at the third layer of the OSI model. Furthermore, with AI/ML/DL tasks continuously being used by many groups for a wide array of tasks, it necessitates innovation for improving performance and connectivity to data sources. With these advancements from NVIDIA, it is expected to only propogate the AI craze even further. As to whether this is a net positive for society is up to debate, however.

3 Dynamic GPU Energy Optimization for Machine Learning Training Workloads

The authors of this publication consist of Farui Wang, Weizhe Zhang, Senior Member IEEE Shichao Lai, Meng Hao, and Zheng Wang.

This paper provides a novel GPU energy optimization framework called GPOEO which is able to dynamically determine the most optimal energy saving configuration to save energy while deploying different machine learning tasks.

As machine learning tasks become longer and greater overtime while processing substantial amounts of data, GPU energy optimization becomes a critical aspect of exploration towards computer architecture and GPUs in general. The author's novel idea of GPOEO utilizes an online model of GPU energy optimization which trades execution time for greater energy efficiency.

The GPOEO framework is comprised of an offline training stage followed by an online optimization stage. During the first stage, the representitive benchmarks are ran on each SM and memory clock frequency in order to find performance counter metrics on the reference SM, memory frequencies, and energy time data. After, the multi objective models are trained. For the second half of the framework, energy and performance counter metrics in a single detected period are measured to predict the best SM and memory frequency. Various frequencies close to the predicted optimal configuration are compared against the measured energy time data to look for the true optimal solution. Lastly, the true optimal solution is set and then its energy characteristics are observed.

The authors reference a few related works which they state as being mostly comprised of being offline models and critique the methods as having a greater learning curve compared to their online counterparts.

The metrics for GPOEO were evaluated using 71 machine learning workloads from two AI benchmark suits which ran on an NVIDIA RTX3080Ti GPU. In comparison to the default NVIDIA scheduling methods, GPOEO performed a mean energy saving of 16.2 percent with an average execution time increase of 5.1 percent.

4 Flexible Instrumentation for Live On-Chip Debug of Machine Learning Training on FPGAs

The authors of this publication consist of Daniel Holanda Noronha, Zhiqiang Que, Wayne Luk and Steven J.E. Wilton.

This paper goes in depth into to research behind FPGA chips and how to improve the training of machine learning tasks by discussing a novel on chip debugging instrumentation.

As FPGAs (Field Programmable Gate Arrays) have shown to be a great option for certain machine learning tasks, the authors of this paper aim to improve upon the issue of FPGAs high computational costs. To do so, monitering on chip data to be able to diagnose any issues in a timely manner is crucial so that the overall cost of training may be reduced. The paper's instrumentation improves on previous works by storing gathered data off chip rather than using up recources of on chip memory. The paper provides motivational examples highlighting the need for clarity regarding run time results of training applications. The main proposition boasts a flexible on chip debugging method for FGPA machine learning training sessions while quantifying the outcomes of utilizing such an infrastructure within hardware accelrators. The study of how these factors impact changes according to certain sets of parameters is also a relevant aspect to this paper.

The paper presents a classification of bugs into five categories: data bugs, syntax bugs, structural bugs, and conceptual bugs. Though previous works have focused on mainly structural bugs during inference, the authors have added onto this concept by incorporating conceptual bugs which only become obvious during training. To provide an example of such conceptual bugs, the authors modeled a DNN with ReLu activation within the hidden layers to execute a classification task. At first the DNN behaves decently during the first few epochs but after the 33rd epoch accuracy starts to slow down significantly but recovers eventually. This instance would not have been found in an initial RTL level simulation. The authors critique current exisiting on chip debugging as being low in flexibility such that most of the solutions on the market lack the ability to dynamically adapt at runtine or perform minimal data compression on chip. The novel chip debugging explained in this paper is superior as the instrumentation generates aggregated date to condense information in a domain specific manner. The proposed system is also firmware programmable which gives programmers and designers the opportunity to receive a broader range of debug information tailored to each specified need towards the training process.

Overall, these four publications discuss different approaches towards improving AI/ML/DL tasks at the hardware level. The first paper introduces a series of three different hardware technologies that are under active research: GPUs, FPGAs, and ASIC chips. These three architectures function very differently. GPUs rely a large amount of simplistic ALUs for processing. FPGAs are reprogrammable logical gate arrays which allow for static creation/setting of the chip. ASIC chips are highly specialized and custom-made chips to function for a particular task. Despite these three hardware approaches being so different, they are capable of improving performance when working on AI tasks. The other three papers discuss how to improve GPUs and FPGAs performance and power efficiency using several methods, both software and hardware.

Before, discussing the various methods discussed to improve performance for these different architectures, we will preface by saying that ASIC chips are almost completely disregarded. This is because they are custom-made to whatever purpose the buyer desires. As a result, they are improved by trying out new hardware techniques created by the buyer themself, not the manufacturer. Thus, any advancements made are not discussed in depth in this summary of the four previously discussed papers.

With that being said, there do exist methods to improve the three interested technologies. GPUs can be improved by utilizing ML techniques, improving hardware connections to network switches for data retrieval/submittal, and using advanced network technologies such as Docker and Kubernetes for large scale deployment. FPGAs can be improved by utilizing ML models to discover bugs while the ML process is running. By using ML, the developers can discover where to improve the architecture layout of the logical gates. A different methodology is to utilize a special type of NN in order to take advantage of the hardware improvements made by the FPGA. ASIC chips instead can be improved by using a novel hardware-software integration for optimal connectivity between the different processing components.

We think that that these technologies are highly desirable espeically given recent advancements in AI with Generative AI models such as ChatGPT, Llama, and Devin. These various improvements prove capable and utilize both software and hardware improvements. By combining these approaches, further advancements towards processing for ML tasks may be discovered.

Overall, these improvements prove extremely useful, espeically given the amount of hype around AI as of recent. There may be additional improvements to improve power efficiency, processing speed, and more for these different hardware architectures. By understanding ML on a hardware level, as well as on a software level, developers may be able to better deliver on public interests.

References

[1] Q. Jiang and J. Zhan, ”Traditional Architecture Artificial Intelligence Chip Technology,” pp. 440-445, Nov. 2021, doi: https://doi.org/10.1109/iccsmt54525.2021.00086.

[2] J. Choquette, ”NVIDIA Hopper H100 GPU: Scaling Performance,” IEEE Micro, pp. 1-13, 2023, doi: https://doi.org/10.1109/mm.2023.3256796.

[3] F. Wang, W. Zhang, S. Lai, M. Hao, and Z. Wang, ”Dynamic GPU Energy Optimization for Machine Learning Training Workloads,” IEEE Transactions on Parallel and Distributed Systems, pp. 1-1, 2022, doi: https://doi.org/10.1109/tpds.2021.3137867.

[4] D. H. Noronha, Z. Que, W. Luk, and Steven, ”Flexible Instrumentation for Live On-Chip Debug of Machine Learning Training on FPGAs,” vol. 3, pp. 20-28, May 2021, doi: https://doi.org/10.1109/fccm51124.2021.00011.