Hardware/Software for Configurable and Reconfigurable Computing of Artificial Intelligence task

# Hena Naaz

# [henanaazkhan24@gmail.com](mailto:henanaazkhan24@gmail.com)

***ABSTRACT***

## *Artificial Intelligence Applications have increased the need of developing efficient architecture for advanced acceleration to accommodate the enormous data load. In this paper, gives an overview of various approach to solving hardware dependency and bottleneck for study and research purposes. The paper proposes that an intelligent architecture should be designed to handle increasing data load well. There are several examples presented that show how to exploit each of the resources described to design a much more efficient and high-performance computing system for study purposes without the need of depending upon per-head development board count for projects and research in the field of accelerated architecture for machine learning applications. The paper also discusses recent research that aims to reduce data transfer overhead and memory latency, practically enabling computations closer to data storage units, with integrated software stack for accelerated performance leading to efficient hardware-software co-design. The paper concludes with some proposal to adopt to limit the dependency on hardware that is also changing frequently to build computing architecture and designs for future study. This short paper provides a summary of the various works that are happening in recent times in both academia as well as industry which can be leveraged to further work that may be beneficial to study the scope of advanced computers.*

1. INTRODUCTION

## Due to huge data over-load, developer and data scientists who are developing Artificial Intelligence applications and machine learning models are in need of computer system designs that can work on data-intensive models to provide results that are accurate and reliable. To achieve the goal of learning efficiently, the hardware design plays an important role as much as the software stack on it. However, with the boom of Artificial Intelligence in the era of technological advancement, the enormous amount of data has exposed the limitation with computing. Computing resources are bottlenecked by data. The emergence of large amounts of data has stresses the storage, data-transfer, and primarily computation capability of the advanced high-end processors that we use. These limitations have therefore impacted the performance optimization process and there is a huge dependency on data center and servers for data communication. As a result, various artificial intelligence application’s performance, efficiency and scalability are bottlenecked by the never-ending data overload.

## The existing computing systems process increasingly large amounts of data, but this infrastructure might not be available for study purposes to researchers because of various license and need of having physical laboratory resources. Data is key for many modern workloads and systems, and this era of digitalization during the Covid-19 pandemic and even furthered the importance of data and need of having resources to extract results out of it.

## Significant workloads (e.g., machine learning, autonomous vehicles, global web broadcast, gene structure analysis), whether they execute on huge data servers or computing systems are all data intensive. All these applications demand efficient processing of large amounts of data with highest level of accuracy possible. With rapid increase in the software industry and data analytics, the system for data-intensive computing is strengthened further with data that is generated more than present day systems can process. The huge overload in data around genes, mutations study and health parameters around the covid-19 surge was a big example of how much there is a need of having secure and reliable computing resources [1]. Processors equipped with units and engines that are specifically computing oriented have been put in place now with the emerging technology in the field of processor architecture. However, the way these Processors are designed, modern computers using them are not efficient at dealing with large amounts of data due to limitation in computing processes that are limited by harware. Therefore, data becomes a large performance and energy bottleneck, and it greatly impacts system reliability and security as well.

There is a need of advanced research in the field of computer architecture in order to develop and collaborate the technological advancements employed in Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Chips (ASIC). To bolster the research community, the dependency on hardware and servers can be overcome using efficient software-hardware co-design approach and tools like simulators and emulators to test on the accuracy of the design. This limits the scope of research due to Electronic Design Automation (EDA) industry license and version maintenance bottlenecks which is suitable more for industry-based projects rather than research projects. If these limitations on research objectives are lifted, then much diversified and robust techniques can be tested at the academic level. This demand was an open-source platform for small scale industry and academics which provides the infrastructure of a configurable and reconfigurable hardware stack which has software coupled optimizations employed to it. As a prime example, there is a research work providing evidence that the potential for new open-source based computer architecture tools and software can prove out to be helpful to develop advanced computer architecture and cross-domain developments like Accelerators for data-intensive applications by using FPGA-cloud without use of giant Servers [2]. Furthermore, the tedious process involved in the design of processor-centric design of modern computing systems is one prime cause of why data overwhelms modern machines.

1. Scope of Hardware

## An intelligent architecture is expected to handle the data storing, accessing, and processing efficiently. What specific hardware architecture can handle data well?

First, the system should ensure that data does not overwhelm its components.

Second, an intelligent architecture takes advantage of the last amounts of data and metadata that flow through the system, to continuously improve its decision making, by bettering its policies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Applications** | **ASIC** | **FPGA** | **CPU** | **GPU** |
| Image Processing |  |  |  |  |
| Search Engines |  |  |  |  |
| Industry controls |  |  |  |  |
| Supercomputer/HPC |  |  |  |  |
| AI Training |  |  |  |  |
| AI Inference |  |  |  |  |
| General-purpose Computing |  |  |  |  |

Table 1*. AI applications across hardware*

1. ASICs and FPGAs

## As per the analysis on the works happening around evolving hardware architectures, it was found that the existing computing architectures are not at-par with the requirement for various computing applications and that’s why they fall short of handling data well. The modern architecture is based on certain hardware to build designs efficiently. Some of them have been analyzed and the observations are stated as follow.

First, modern architectures are poor at dealing with data: they are designed to mainly store and move data, as opposed to compute on the data. Most system resources serve the processor (and accelerators) without being capable of processing data. Doing so would eliminate the huge data access bottleneck of processor-centric systems, thereby improving performance, reducing energy consumption, alleviating off-chip bandwidth requirements (and hence area and cost), likely reducing system and hardware design complexity, as well as opening new opportunities for improving system security and reliability by handling data more locally in or near where it resides.

Second, classical architectures are not proficient enough to take advantage of the data available to them during online operation and over time. Because the models that they are employing are still using same hardware architectures that are designed with a policy that is rigid and hardcoded by a developer, who might not be aware about the use cases most suitable for on-fly runs and operations. To achieve exceptional performance, it is important to stress the available memory and make use of the hardware resources as judiciously as possible. The reconfigurable technology like FPGAs too have this demerit when the design is synthesized and put on the available Configurable Logic Blocks using Look Up Tables.

I propose a design which can be not just reconfigurable but also can provide modularity at the level of parameterized design to leverage the hardware resources. To choose the best design, there is a need of adopting methodology to divide the portion of FPGA such that a small portion is reserved for employing security checks and rest are meant for calculation of data with a place and route policy that improves the throughput, that I believe will provide the highest benefits in the future if design cones are made to identify portion of designs having most amount of data communication and those data can be stored temporarily in the on chip memory for better performances. This calls for a pseudo-cache memory to be mindful of design dependency by integrating a customizable block focused at maintaining the design dependency cone. To design such methodologies, proposals, and projects like usage of FireSim FPGA-accelerated simulation can be of great importance. Furthermore, for accelerated deterministic simulations there is still a need for automation around the interactions between the host machine and the FPGA, that will make the FireSim users free from directly interacting with the FPGA toolchain and the FPGA-specific configurations [2]. These simulators can thus provide the infrastructure which can be configurable and reconfigurable. To achieve the goal of automating the host and FPGA interactions, various hardware specific transactors or Ips can be leveraged to give more programmability to user through mere use of a laptop to specify tasks. To design architecture intelligence and far-sightedness in controller and system policies in an architecture is necessary for obtaining good performance and efficiency (as well as better reliability, security, and perhaps other metrics) under a variety of system conditions and workloads.

There are application-specific integrated circuits chips in the industry that are manufactured for a specific AI task as well, but ASICs lack the flexibility of FPGAs and can’t be reprogrammed. Proposed usage encourages use of design having configurable and reconfigurable, modern architectures are poor at knowing and exploiting different properties of application and system data. If the characteristics of the data to be accessed or manipulated were known, the decisions taken could be very different, for maximizing both performance and efficiency. FPGA stacks can be programmed for custom computing engines dedicated to running machine learning and neural network models. Thus, applying machine learning models to understand the data elements through compiler would further make the design more reliable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Processing (single)**  **TFLOPs** | **Power Utilization**  **GFLOPs/W** | **Cost**  **$** |
| Nvidia GeForce GT 730 | 0.69 | 7 |  |
| Radeon R9 390x | 5.91 | 18 |  |
| Radeon R9 FuryX | 7.17 | 20 |  |
| Artix-7 200T | 0.65 | 72 |  |
| Kintex-7 480T | 1.80 | 72 |  |
| Virtex-7 690T | 3.12 | 78 |  |

*Table 2. Key Performance Indicators: GPUs and FPGAs*

1. GPUs and Accelerators

## Accelerators for machine learning applications are now making use of GPU due to its matrix calculation capability for large amount of data being processed on similar functions. This paper does not go into detail but provides a brief overview with references to other works using such architectures.

*DNN*

A DNN Accelerator interacts frequently with the off-chip memory and RAM for data operations, and communication through the PE arrays. By mapping PE arrays on various compute-pipes on GPUs, it adds to the overhead of long data path and race-around that can arise due to continuous flow of data over fast data slow. This can be however, achieved with efficient place-and-route to reduce the data communication overhead and this will also reduce the chance of data corruption due to overwritten malicious data coming from different compute lanes or fault injections. Recent results show up to approximately two orders of magnitude improvement in energy and performance over conventional processor-centric systems.

*CNN*

In other words, it customizes itself (i.e., its policies and mechanisms) to the characteristics of the data and computations it is dealing with. Such an architecture requires knowledge of various characteristics of different data elements and structures as well as computations.

However, obtaining fast and reliable outcome for Artificial Intelligence applications on GPUs has its own limitations. When a chip is made purposefully for a given deep learning workload, i.e., ASICs, it offers better performance as compared to general purpose chips like GPUs. Additionally, FPGAs are hardware that can be customized with integrated AI and can be programmed to deliver task as expected from a GPU or an ASIC.

Chart, bar chart

Description automatically generated

*Figure 2. Comparison between GPUs and FPGAs performance*

1. CONCLUSION

## This paper compares the key performance indicators of hardware like GPUs and FPGAs, for Artificial Intelligence applications involving processing and compute over large data loads. The analysis also compared cost-efficient GPUs with big floating point processing capacity, and power efficiency. In the future, for a safe and reliable system which has configurable and reconfigurable units can consist of GPUs with improved power efficiency, and FPGAs with better compute power, low cost, and easy maintenance. In order to accelerate the performance for Artificial intelligence training and inference, the GPU based multi-instance architecture can be leveraged and for inference low cost and reconfigurable FPGA units can be combined together like a SoC for reduced latency and can be scaled further for better throughput. The architecture can be further made user-friendly through optimized on-fly microarchitecture that is closely coupled with software stack for highest programmability and control.

ACKNOWLEDGEMENTS

## This paper aims at delivering the prospective research area in the field of hardware design for data intensive applications in the field of Machine Learning or Artificial Intelligence Hardware Systems. I am thankful to all the researchers in academic and industrial domain who have worked directly or indirectly to contribute towards increasing the computation capability of the systems with ever-increasing data-load in the age of data, and the references mentioned in the paper.

REFERENCES

1. J. s. Emer and D. W. Clark, “A characterization of processor performance”, ACM 1984
2. Krste Asavonic, Yakun Sophia Shao, Borivoje Nikolic, “Vertically Integrated Computing Labs Using Open-source Hardware Generators and Cloud-Hostel FPGAs”, IEEE 2021
3. Onur Mutlu, “Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU)”, IEEE
4. Z. D. Stevens et al., “Big data: astronomical or genomical?”, PLoS Biology, 2015.
5. S. Ghose et al., “Processing-in-Memory: A Workload-Driven Perspective”, IBM JRD 2019.
6. O. Mutlu et al., “Processing Data Where It Makes Sense: Enabling In- Memory Computation”, MICPRO, 2019.
7. A. Boroumand et al., “Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks”, ASPLOS 2018.
8. O. Mutlu, “Enabling Computation with Minimal Data Movement: Changing the Computing Paradigm for High Efficiency", Design Automation Summer School Lecture, DAC 2019.
9. J. Ahn et al., “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing”, ISCA 2015.
10. BWP001 v1.0 “GPU vs FPGA Performance Comparison”, Berten Digital Signal Processing, 2016