# Accelerate your AI Cloud infrastructure: a virtualization perspective

Liang Yan <lyan@suse.com>

***Keywords***— KVM, Hardware acceleration, AI, FPGA Virtualization

## 1 Introduction

In recent years, the popularity and the success of the Internet, especially mobile network, computing resources have become cheaper, more powerful and more available than ever before , this technological trend is known as cloud computing.

The cloud computing has gone through different stages since beginning. It is attributed to platforms at first, for instance, Google App Engine, Heroku, Azure all delivered Platform as a Service (PaaS) to customers. The next big thing in the cloud was Infrastructure as a Service where customers could manage virtual machines and all the resources by themselves. Then, cloud was centered to data. From relational databases to big data to graph databases, cloud providers different kinds of data platform services.

Today, the next step that would drive the growth of public cloud would be artificial intelligence. Cloud providers are gearing up to offer a comprehensive stack that provide AI as a Service. AI and Cloud Computing together start to reshape the IT Infrastructure. Often combined with GPUs, FPGAs, and other optimized ASICs, the AI hardware architectures act as a substantial workload within many application platforms

Cloud makes building AI applications enticingly easy. Since most companies struggle to find the right skills to start an AI project, this becomes quite attractive. Also, Cloud offers ease of use, even click-and-go simplicity in a area full of relatively obscure technology. After all, elastic cloud services can offer a flexible hardware infrastructure to AI, with state-of-the-art GPUs or FPGAs to accelerate the training process and handle the inference process. AI Developers don't have to deal with complex hardware configuration and purchase decisions, the AI software stacks and development frameworks are all ready to go[1].

A simple framework setup in cloud environment is easy, however binding hardware accelerator to cloud is totally a different story. Hardware accelerators are usually used as dedicated resources, a lot of things should be considered when integrated into cloud. This paper will present a general implementation for all of

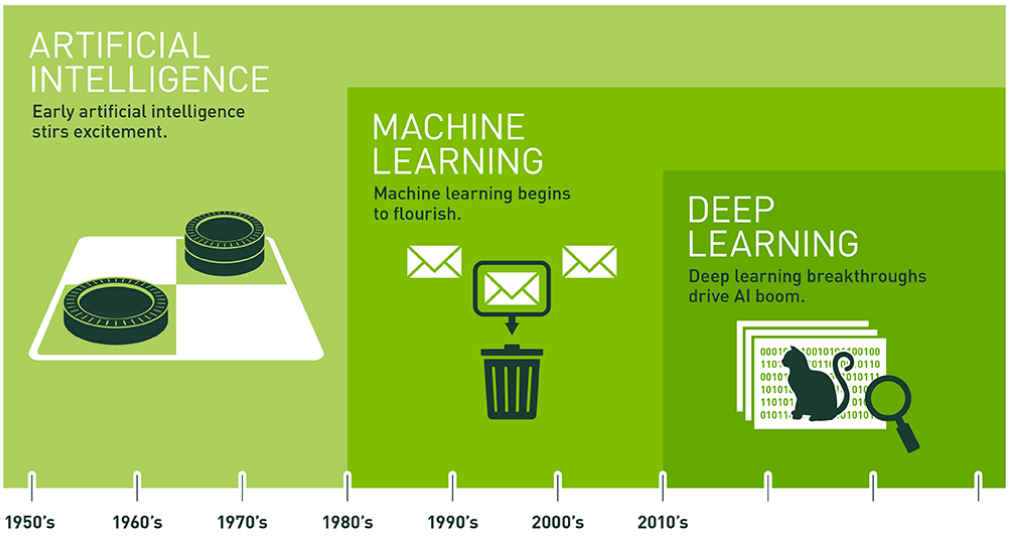


Figure 1: AI vs ML vs DL [2]

them. First, we will have a close look on AI/ML/DL, hardware accelerator, and AI cloud. Then, we will focus on virtualization implementation for each of them. Last, we will see how these hardware accelerators work with a VM.

# 2 Background

## 2.1 Artificial Intelligence vs Machine Learning vs Deep Learning

As is shown in Figure 1, ML(Machine Learning) is a subset of AI(Artificial Intelligence), and DL(Deep Learning) is a subset of ML[3].

AI was first introduced in the 1950s, it is a much broader concept than ML and DL. It usually refers to the capability of machines to carry out "intellectual" algorithms. It separated into several areas since then, ML is one of the important ones which begins 1980s. ML is actually a training algorithm by analysing a lot of data. DL is simply a technique for realizing machine learning. In other words, DL is a specific implementation of machine learning.

DL algorithms are basically inspired by the information processing patterns found from the human brain. Just like we use our brains to identify patterns and classify various types of information, DL algorithms can be trained to accomplish the same tasks for machines.

While ML uses algorithms to parse and learn from data, to make informed decisions, DL structures algorithms to create an convolution neural network (CNN) that can learn, and similar to human intelligence, can make accurate decisions

on its own. Therefore, instead of designing algorithms by hand, systems can be trained to implement algorithms in a way similar to what comes to humans, and sometimes even exceeding human-level performance.

## 2.2 Training and Inference[4]

At a high level, the goal of AI is to replace algorithms that programmed by people with the ones that could adapt on their own. For example, DL is used to learn features and patterns that best represent data automatically. This process is called Training. Training is the phase in which the computer essentially tries to learn from your data.

Inference happens after training and only could happen after that. The goal is to learn how to do a job just like a human brain. This job phase is called inference, regardless if the trained neural network learned how to recognize images, text, or any other things. Inference essentially takes real-world data and response with a prediction result quickly.

Training computers to think like humans is achieved through the use of neural networks in DL. Neural networks are a series of algorithms modeled after the human brain. Neural networks can recognize patterns and categorize and classify information, then they label and assign items to different categories.

Today, GPU and TPU are mainly used for training while FPGAs and ASICs are used for Inference.

## 2.3 Hardware Accelerator[5]

Think of convolution neural network(CNN) as the main algorithm for deep learning. To provide more accurate results and faster object recognition, the CNN needs to add more layers. However, more NN layers results in more complex CNN structures as well as higher depth of CNN models. Thus, tons of operations and parameters, as well as substantial computing resources will be required to train and evaluate this large scale CNN. Such requirements represent a computational challenge for general purpose processors (GPP). Consequently, hardware accelerators such as application specific integrated circuit (ASIC), field programmable gate array (FPGA), and graphic processing unit (GPU) have been employed to improve the CNN performance.

GPUs are the most widely used hardware accelerators for improving both training and inference in CNNs. This is due to the high memory bandwidth and throughput as well as highly efficient in floating-point operations. However, GPU accelerators consume a large amount of power. Furthermore, GPUs leverage their performance from the ability to process a large batch in parallel, will not be good for application that run sequentially, such as video stream, where input images should be processed frame by frame.

FPGA and ASIC hardware accelerators have relatively limited memory, I/O

bandwidths, and computing resources compared with GPUs. However, they can achieve at least moderate performance with lower power consumption. The throughput of ASIC design can be improved by customizing memory hierarchy and assigning dedicated resources. However, the development cycle, cost, and flexibility are not good for ASIC-based acceleration of deep learning networks. Alternatively, FPGA-based accelerators are currently in use to provide high throughput at a reasonable cost with low power consumption and good flexibility.

### 2.4 AI Cloud[6]

A lot of AI development, especially training deep neural networks, demands massive computation. Also, we don't stop training a network in order to keep it fresh with new data and features, or even build a completely new network to improve accuracy with a new algorithms. In this case, a cloud environment would be more convent for management. Actually, in a real word, cloud is necessary for AI framework for two main reasons: First, only a cloud platform could provide enough data for AI process; Second, cloud platform could provide a scale data services with a moderate expense. A company may know how they need to utilize AI, they just don't have the methods to build an application or algorithms to get the outcomes they want.

Some startups have already established this combination of AI and Cloud to flourish in business. Veritone has built up an AI operating system utilizing a cloud-based cognitive computing system which uses huge amounts of datasets from various sources. Another example is of Quantifi, which uses AI and machine learning analysis to improve digital advert placements for brands. Quantifi customers can pull out the power of data which has been gathered from a large number of other digital ad tests. This would not be implemented without the cloud.

As the growth of cloud computing has enabled the AI area to prosper, AI development also starts to drive the cloud business forward. Throughout the next couple years, we can expect the business keeps going up with AI driving cloud computing higher than ever, while the cloud business proves the advantages of AI power than before.

## 3 GPU Virtualization

GPU virtualization means vGPU could have full features for each VM, and VMs do not need to change driver for library compatible, and most importantly it could still have high performance[7]. Three major vendors all have implemented Full virtualization today.

AMD provides Multi-user GPU with SRIOV (Single Root I/O virtualization) based GPU virtualization[8].

Nvidia GRID technology uses a vGPU manager to share GPU based on time slices. The graphics commands of each VM are passed directly to the GPU[9].

KVMGT is the implementation of Intel GVT-g technology, a full GPU virtualization solution. Under Intel GVT-g, a virtual GPU instance is maintained for each VM, with part of performance critical resources directly passthrough.

The key difference between them is how handles VRAM. AMD's MxGPU is 100% hardware-based, the individual virtual machines framebuffers are physically isolated from one another, whereas with NVIDIA and Intel, the isolation is done by software. AMD differs from the others in how it slices up the shader engines, too. With MxGPU, virtual machines get a dedicated, physical slice of the shaders, whereas Intel and NVIDIA time slice VMs across all of their shaders. The difference is that time slicing gives the users 100% of the GPU for a proportional amount of time, whereas AMD gives users a proportional amount of GPU 100% of the time[9].

So If Intel or NVIDIA GPU users aren't using the GPU, that frees up more longer slices of time for users that are using the GPU, the fewer the users the better the performance. For AMD, if a user isn't using the GPU, those dedicated shaders will be unused. That said, since the slicing NVIDIA and Intel is done in software, the GPU can't be turned back over to the pool until the last command has finished executing. That means the misbehaving applications could starve other VMs of GPU resources.

# 4 ASICs for AI

Application-Specific Integrated Circuits(ASICs) are actually hardware implementation for software algorithms. Technically, a GPU is an ASIC used for graphics processing algorithms. ASIC offers an instruction set and libraries to allow the GPU to be programmed as an accelerator for many parallel algorithms. GPUs are good at performing matrix operations that underlie graphics, AI and many scientific algorithms. Basically, GPUs are very fast and relatively flexible.

The alternative is to design a custom ASIC dedicated to perform fixed operations extremely fast since the entire chip's logic area can be focus to a set of narrow functions. In the case of the Google TPU[10], they lend themselves well to a high degree of parallelism, and processing neural networks as an "embarrassingly parallel" workload. ASIC can go very fast, but it can only good for some specific functions.

All of the ASICs provide some sort of acceleration to run a deep learning framework. Nervana developed its own framework called Neon, whereas Graphcore and Wave Computing both support TensorFlow. The ASIC companies are also developing their own boards and nodes that can be plugged into servers with minimal modifications. Application developers can write a deep learning algorithm, set some compile time options, and continue to develop software just as they would on the CPU platform, completely oblivious to the underlying hardware.

By the way, there are no full virtualization for ASICs so far, PCIe pass-through are mainly used by VM.

# 5 FPGA Virtualization

Field programmable gate arrays (FPGAs) are commercial devices that can be programmed to implement custom digital circuits. Fundamentally, they consist of a mesh of basic circuit resources that can be configured to complex architectures. The whole architecture is described in a hardware description language (HDL), or high level algorithmic code, and automated tools work out how to build it using the components in the FPGA, how they should be arranged on the grid of the FPGA and connected, finally generating a bitstream — a binary file which is loaded into the FPGA to implement the circuit.

FPGAs have also been explored more recently. Developments are easing the integration of FPGAs in the datacenter. At the server hardware level, IBM's POWER8 Coherent Accelerator Processor Interface (CAPI) allows tighter coupling between the processor and a co-processing peripheral. Intel's XEON+FPGA integrates an FPGA with a XEON processor in a single chip package, and their recent purchase of Altera implys the importance of hardware accelerators in the data-center area. Microsoft also presented a demonstration on using FPGAs in the data-center for Bing search algorithm.

## 5.1 Classification[11]

Above proofs show using FPGAs as a general cloud computing resource is a serious case. However, FPGAs have only been used as static accelerators so far, designed once and used for a single function. In the cloud environment, the ability to modify accelerator functions at runtime, FPGA virtualization, becomes more important.

The term "Virtualization" for FPGA, has changed over time due to the change in application requirements when compared to an earlier survey on FPGA virtualization in 2004 by Plessl and Platzner[12]. In the survey, FPGA virtualization was classified into three categories: temporal partitioning, virtualized execution, and virtual machine.

Temporal partitioning is used to fit large designs for relatively smaller FPGAs by reconfiguring an FPGA to host a design partition at a time. This was the first FPGA virtualization approach when device capacity was often not sufficient to host.

Virtualized execution is used to define the approach of splitting applications into multiple communicating tasks and using a run-time system to manage them. The aim of this was to support device independence within a device family.

Last, Plessl and Platzner defined virtual machines to provide complete device independence by using an abstract architecture to describe applications. This architecture could be translated later into native architecture by a remapping tool or an interpreter. This approach, in particular, now falls under Overlays. The term "virtual machine" is used for the static architecture these days which provides support for accelerators and is often referred to as a Shell or Hypervisor

for virtual FPGAs.

Nowadays, FPGA virtualization is starting to integrated with software virtualization at a conceptual level, with growing support for heterogeneous systems and concepts such as Acceleration as a Service (AaaS).

# 6 Hardware acceleration in Virtualization Stack

Beside the resource management layer on the host side, direct I/O virtualization is the main mechanism for hardware accelerator inside a vm. It provides a hardware mechanisms for building a virtualized environment with complete device data transfer isolation, different implementations have been provided, such as Intel VT-d [13], AMD IOMMU [14], and PCI-SIG IOV[15]. VT-d and IOMMU are similar. They ensure the isolation I/O address space between different VMs. An I/O MMU similar to MMU installed on PCI bridge to map DMA address to machine memory address. And an IOTLB accelerate this translation. PCI-SIG IOV includes ATS (Address Translation Services), SR-IOV (Single Root IOV), MR-IOV (Multi-Root IOV)[8].

## 6.1 VFIO Platform

The VFIO driver framework provides unified APIs for direct device access in user-space level. It makes full usage of DMA remapping and Interrupt remapping, exposes direct device access to user space in a secure, IOMMU-protected environment. It provides a low latency and high bandwidth access for guest drivers.[16]

## 6.2 MDEV Platform[17]

MDEV platform is a platform device for the devices that do not have built-in SR-IOV capability. It provides a unified management interface for such devices by identifying common requirements. This framework reuses vfio access mechanism, and it could be used for multiple devices, such as GPUs, network adapters, and compute accelerators.

The mediated core driver provides a common interface for mediated device management that can be used by drivers of different devices(Figure 2). This module provides a generic interface to perform these operations:

Create and destroy a mediated device

Add a mediated device to and remove it from a mediated bus driver

Add a mediated device to and remove it from an IOMMU group

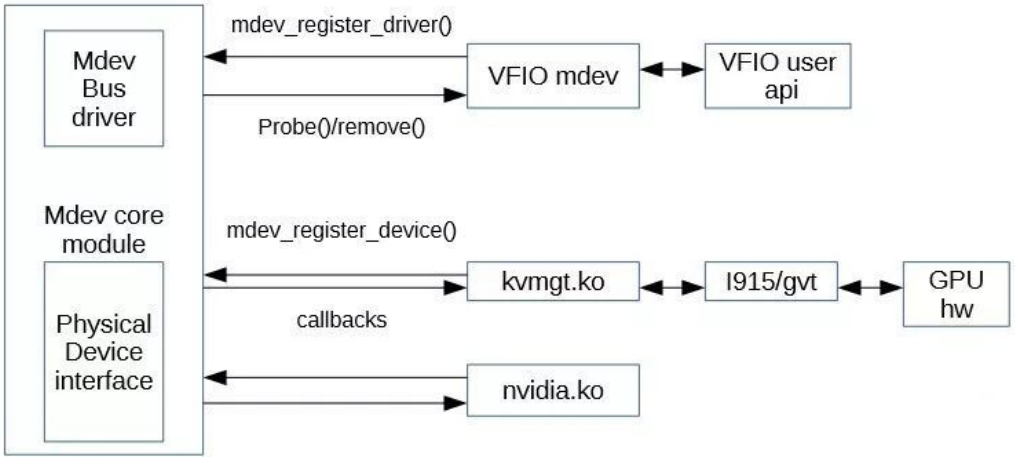


Figure 2: Mdev Platform Overview

# 7 Conclusion

In this paper, we reviewed the recent developments for different hardware accelerators in AI cloud. GPU is still the dominator for deep learning, especially training process, however FPGA and ASIC start to catch up, some ASICs like TPU have already provided impressive performance for some DL algorithms. With the development of hardware acceleration for virtualization, We expect AI and Cloud achieve a new level.

# References

[1] D. Puthal, B. P. S. Sahoo, S. Mishra, and S. Swain. Cloud computing features, issues, and challenges: A big picture. In *2015 International Conference on Computational Intelligence and Networks*, pages 116–123, Jan 2015.

[2] Venkatesan M. Artificial intelligence vs. machine learning vs. deep learning, 2018.

[3] Adnan Raja. How is the cloud enabling artificial intelligence?, 2018.

[4] Daniel. What's the difference between machine learning training and inference?, 2017.

[5] Tim Hwang. Computational power and the social impact of artificial intelligence. *SSRN Electronic Journal*, 03 2018.

[6] Priya Dialani. The fusion of artificial intelligence and cloud computing, 2018.

[7] David Ott. Api remoting turnkey solution for virtualizing hardware resources, 2016.

[8] Y. Dong, X. Yang, X. Li, J. Li, K. Tian, and H. Guan. High performance network virtualization with sr-iov. In *HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*, pages 1–10, Jan 2010.

[9] Kun Tian, Yaozu Dong, and David Cowperthwaite. A full GPU virtualization solution with mediated pass-through. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 121–132, Philadelphia, PA, 2014. USENIX Association.

[10] Kaz Sato, Cliff Young, and David Patterson. An in-depth look at google's first tensor processing unit (tpu), 2017.

[11] A. Shawahna, S. M. Sait, and A. El-Maleh. Fpga-based accelerators of deep learning networks for learning and classification: A review. *IEEE Access*, 7:7823–7859, 2019.

[12] Anuj Vaishnav, Khoa Pham, and Dirk Koch. A survey on fpga virtualization. 09 2018.

[13] Gabe Knuth. Understanding vt-d: Intel virtualization technology for directed i/o, 2009.

[14] Inc Advanced Micro Devices. Adm i/o virtualization technology (iommu) specification, 2016.

[15] PCI-SIG. I/o virtualization., 2007.

[16] Linux Kernel. Vfio - "virtual function i/o", 2018.

[17] Linux Kernel. Add mediated device support, 2016.