FPGAs in The Cloud

Miriam Leeser[1], Suranga Handagala[1], and Michael Zink[1]

[1]Affiliation not available

November 9, 2021

Abstract

As cloud computing grows, the types of computational hardware available in the cloud are diversifying. Field Programmable Gate Arrays (FPGAs) are a relatively new addition to high-performance computing in the cloud, with the ability to accelerate a range of different applications, and the flexibility to offer different cloud computing models. A new and growing configuration is to have the FPGAs directly connected to the network and thus reduce the latency in delivering data to processing elements. We survey the state-of-the-art in FPGAs in the cloud and present the Open Cloud Testbed (OCT), a testbed for research and experimentation into new cloud platforms, which includes network-attached FPGAs in the cloud.

Introduction

Cloud computing has changed the way we develop, release and consume software. Cloud computing includes High-Performance Computing (HPC) systems, and these systems are becoming increasingly heterogeneous. Graphics Processing Units (GPUs) have been widely available in HPC systems for some time, and the Top500 list of supercomputers includes many systems that benefit from GPU acceleration. Field Programmable Gate Arrays (FPGAs) are another heterogeneous architecture and are emerging as popular accelerators in cloud and HPC systems. FPGAs are more flexible than GPUs and can be adapted by the user to solve a host of different problems. Their flexibility means that they often consume less power than alternative processors. It also comes at a cost, as FPGAs can be more challenging for users to program. Applications where FPGAs are popularly deployed include genomics and molecular dynamics, video and image processing, machine learning and data analytics, and in-network data processing.

Cloud computing offers users computing resources that they can access remotely. These resources are made available following several different models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). With IaaS, the base infrastructure is provided for the user, including servers, storage and networking. PaaS adds middleware and development tools to IaaS. In the SaaS model, software solutions are provided for the user, and in many cases, the user is not aware what infrastructure is being used to support their programming problem.

As FPGAs are added to cloud computing systems, it is not yet clear whether they fit into one of these existing cloud computing modes, or will be offered using a new model, FPGA as a Service, FaaS, and indeed what FaaS would mean.

Background: Existing FPGA Offerings in the Cloud

Two existing offerings of FPGAs by cloud service providers take different approaches to how FPGAs should be made available to users. Amazon Web Services (AWS) provide FPGAs as part of their F1 instances. This

follows the PaaS model where users can request F1 instances and run their code directly on these FPGAs using the infrastructure provided by Amazon. Microsoft provides FPGAs in the cloud following the SaaS model as part of Project Catapult. Microsoft Azure applications and Bing searches may be accelerated using FPGAs without the user being aware what processing units are used. The implementations are done by Microsoft engineers and provided as a service.

Hardware models for FPGA in the Cloud

The traditional architecture for computer systems with attached accelerators is a server centric model where the accelerator is attached to a host processor, typically over high-speed interconnect such as PCIe. Such an architecture incurs a high cost, where data has to be transferred first to the host, and then to the accelerator, before it is processed and transferred back to the host. While the accelerator can significantly reduce processing time, the overhead of data transfer may result in little or no end-to-end improvement for an application.

An emerging alternative is to connect accelerators including GPUs and FPGAs directly to the network. In this scenario, the host processor may be responsible to program the FPGA with a bitstream, but the data flow and processing is all direct to the FPGA. This is part of a larger trend of disaggregation of the data center, where memories and other components are generally available, and no longer tied to a server. Microsoft uses this model by making the FPGA part of a SmartNIC to accelerate Azure applications [1]. A similar model is available through the OpenCloud Testbed.

The OpenCloud Testbed

The OpenCloud Testbed (OCT) [2] is part of an NSF CISE Community Research Infrastructure (CRI) grant that includes FPGAs in the cloud. OCT builds on three existing systems: The Massachusetts Green High Performance Computing Center (MGHPCC), The Massachusetts Open Cloud (MOC) and CloudLab, and is available to any US researcher.

The MGHPCC is a facility in Holyoke, MA, for housing research computing systems from a five-university consortium (University of Massachusetts, Harvard University, Boston University, Northeastern University, and the Massachusetts Institute of Technology). MGHPCC provides the space, power, and cooling capacity for approximately 750 racks of computing equipment on a single shared floor.

The Mass Open Cloud (MOC) is a best-effort cloud developed by a partnership of the same five academic institutions with government (Mass Tech Collaborative, USAF), and industry (Red Hat, Intel, Two Sigma, NetApp, Cisco). The existing MOC physical infrastructure includes around 2200 cores of commodity Intel compute, 160 Power9 cores, 40 GPUs, and 1.2PB of storage. The MOC was designed as both a research and a production cloud, where researchers can obtain metrics from running cloud applications and use the information to enhance their tools and modeling for cloud workflows.

CloudLab [3] is particularly aimed at cloud researchers, and provides them with control and visibility all the way down to bare metal. A researcher can provision an entire cloud inside of CloudLab. Most CloudLab resources provide hard isolation from other users, so it can support hundreds of simultaneous "slices," with each getting an artifact-free environment suitable for scientific experimentation with new cloud architectures. A researcher can run standard cloud software stacks such as OpenStack and Hadoop, or build their own from the ground up.

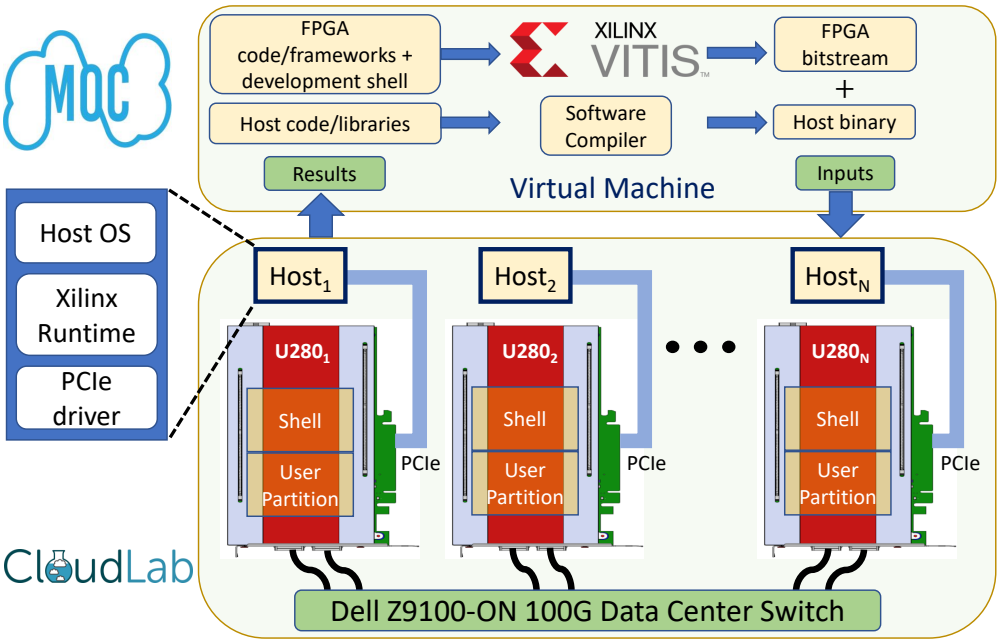

Figure 1: The Open Cloud Testbed: Hardware and software

The Open Cloud Testbed builds on these three projects. Hardware is located in the MGHPCC. A researcher runs the Vitis FPGA development tools in the MOC, and then requests hardware from CloudLab to deploy and run their experiments, as shown in Figure 1. OCT provides FPGAs to the user. There are currently 8 Xilinx Alveo U280s available, each connected to its own host via PCIe and each with two 100 Gbit/s connections directly to the data switch using QSFP28 transceivers. Eight more will be deployed in the near future.

What is different about OCT? It gives users direct access to FPGAs in the cloud and allows them to experiment with the entire setup, including the OS. While development tools are provided, the deployed system is bare metal, giving the user complete freedom to conduct their research. FPGAs are directly connected to the network, enabling direct FPGA-to-FPGA connections and smart NIC based experiments. FPGAs can communicate with the network using either the TCP/IP stack provided by ETH Zürich, or the UDP stack provided by Xilinx. The OCT model is a combination of PaaS and IaaS, where infrastructure in the form of tools for development are made available, and coupled with bare metal deployment. This FPGA offering is much more flexible than that available from AWS, which does not include network attached FPGAs, or from the Microsoft Catapult project, who does not allow users to directly program the FPGAs.

Initial results show the advantage of direct network FPGA-to-FPGA communications. Using an example benchmark from Xilinx, the measured round-trip time (RTT) is around 1 microsecond. The RTT is measured by starting a counter just before the packet is sent by FPGA 1, and stopping it once the same packet is received back at FPGA 1 after it being sent to FPGA 2. We also measured the OpenCL kernel execution time for an application running between two FPGAs where the communication goes from Host 1 to FPGA 1 over PCIe, then FPGA 1 to FPGA 2 over the network. The packet is then looped back from FPGA 2 to FPGA 1 which receives the packet and transmits it back to Host 1. The kernel execution time between sending and receiving a packet was observed to be between 200˜300 microseconds. The high latency in this case is due to OpenCL function calls and the overhead of going through the PCIe connection. The direct FPGA-to-FPGA communication is 1 microsecond. These results argue for disaggregating FPGAs and

host computers and treating the FPGA as a first-class citizen in the cloud.

OCT documentation is available on GitHub where we host getting-started tutorials for both MOC and CloudLab. These tutorials demonstrate the workflow for stand-alone and network-attached accelerator development and deployment from the ground up.

Security in the Cloud

The bare-metal approach used in OCT gives the research community maximum freedom for system experimentation and evaluation but also bears certain risks when it comes to security. The provision of bare-metal servers gives users access to all components of the system, which means that they can also compromise (by accident or on purpose) the firmware of the system. Such modifications can impact the security of the system and subsequent users will work on a compromised system without being aware of it. OCT uses the mechanisms of Elastic Secure Infrastructure (ESI) [4] and its attestation service to provide an uncompromised system and make it available to an experimenter. Currently, ESI provides this service only for the servers that house the FPGAs in OCT. To ensure that a new user receives an uncompromised FPGA with the start of every new lease of a bare-metal system, we enforce the execution of a procedure that is automatically performed at the startup of the host server. This procedure makes sure the FPGA is put into a known state and no information accidentally or intentionally left behind from an earlier user continues to reside on it. An official Xilinx Run Time (XRT) is installed as is the hardware shell. The shell is only reinstalled if the new user wishes to change the version, however all previous user logic is removed. In addition, the network that the FPGAs directly connect to is isolated to guarantee that a networked FPGA cannot inject spurious packets into a production network.

Example Application: Machine Learning on FPGAs in the Cloud

We provide sample applications including the FINN framework for machine learning. FINN [5] is developed and maintained by Xilinx Research Labs to explore deep neural network inference on FPGAs. The FINN compiler is used to create data-flow architectures that can be parallelized across and within different layers of a neural network, and transform the data-flow architecture to a bit file that can be run on FPGA hardware.

With the amount of resources available in the OCT, we are particularly interested in implementing network-attached FINN accelerators split across multiple FPGAs with convolutional neural network types such as MobileNet and ResNet, whose partitioning is discussed by Alonso at al. [6]. Figure 2 shows an arrangement of this where MobileNet is implemented with three accelerators that are mapped to two FPGAs. Two of the accelerators function stand-alone, while the third, which contains all the communications required between the FPGAs, is split between two Xilinx U280s. The two halves of this accelerator are connected using the network infrastructure of the UDP stack, which enables communication between them via the switch.

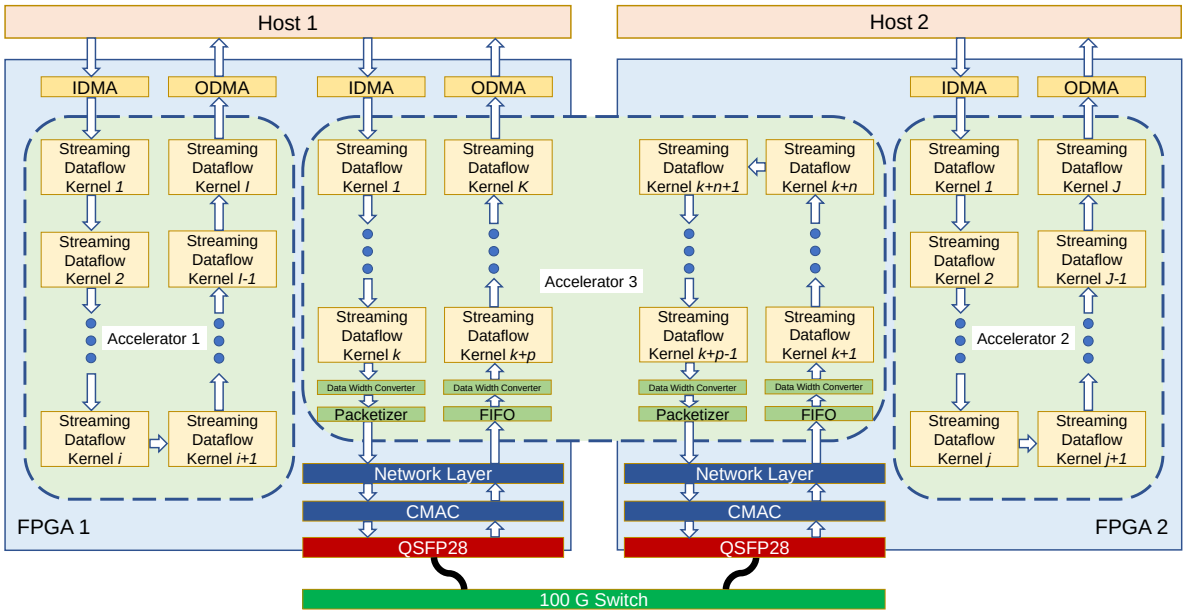

Figure 2: Three MobileNet accelerators implemented using two U280 accelerator cards

The Future

The future of high-performance and scientific computing is in the cloud, with accelerators including FPGAs providing reconfigurable hardware that can be adapted to a user's applications. The trend is to replace server-centric architectures with a disaggregated data center, where memory and accelerators are connect directly to the network. Disaggregation removes many existing bottlenecks and ensures better utilization of resources. It also opens up new research directions, including assigning and scheduling jobs to processors, be they CPUs, GPUs or FPGAs as well as writing applications that target each type of accelerator. Security is also a challenge when there are many devices directly connected to the network and to one another. Our research aims to make FPGAs in the cloud available to a large number of researchers. Cloud users of the future will be able to take advantage of a computer platform with fewer memory bottlenecks and the processing power to deliver graph processing, machine learning, security and privacy applications on large data as well as accelerating scientific applications and new applications not yet imagined. With OCT, systems researchers have a platform where they can provide FPGAs as a Service (FaaS) and experiment with what that should look like.

Acknowledgements

Open Cloud Testbed (OCT) is funded by by National Science Foundation grant CNS-1925658, a CISE Computer Research Infrastructure award. The authors would like to thank the many researchers involved with OCT, especially Peter Desnoyers, Martin Herbordt, Orran Krieger and David Irwin. We would also like to thank the industry partners of MOC, and OCT, and especially Xilinx Inc. for their generous donations to OCT in support of this project.

Miriam Leeser is a Professor in Electrical and Computer Engineering at Northeastern where she is head of the Reconfigurable Computing Laboratory. She is a member of the Computer Engineering research group. Her research includes using heterogeneous architectures from the cloud to the edge with applications in data privacy and wireless communications. She is a senior member of the IEEE and of the ACM. Contact her at mel@coe.neu.edu.

Suranga Handagala is an FPGA Engineer at Khoury College of Computer Sciences. He has a PhD in Electrical Engineering from Northeastern University. Contact him at s.handagala@northeastern.edu.

Michael Zink is a Professor of Electrical and Computer Engineering at UMass Amherst. He is a member of the Computer Engineering group that focuses on internet and sensing systems. His research focuses on Cyberinfrastructure, IoT, and Multimedia Systems. He is a senior member of the IEEE and the ACM. Contact him at zink@ecs.umass.edu.

References

[1]D. Firestone *et al.*, "Azure accelerated networking: SmartNICs in the public cloud", in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI*18), 2018.

[2]M. Zink et al., "The Open Cloud Testbed (OCT): A Platform for Research into new Cloud Technologies", in IEEE CloudNet, 2021.

[3]"The design and operation of CloudLab", in USENIX Annual Technical Conference, 2019.

[4]A. Mosayyebzadeh et al., "A secure cloud with minimal provider trust", in 10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18), 2018.

[5]M. Blott et al., "FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks", ACM Transactions on Reconfigurable Technology and Systems, vol. 11, no. 3, pp. 1–23, Dec. 2018, doi: 10.1145/3242897.

[6]T. Alonso et al., "Elastic-DF: Scaling Performance of DNN Inference in FPGA Clouds through Automatic Partitioning", 2021.