© 2017 IEEE *

An FPGA Platform for Hyperscalers

F. Abel, J. Weerasinghe, C. Hagleitner, B. Weiss, S. Paredes
IBM Research – Zurich
Säumerstrasse 4, 8803 Rüschlikon, Switzerland
{fab, wee, hle, wei, spa}@zurich.ibm.com

***Abstract*—FPGAs (Field Programmable Gate Arrays) are making their way into data centers (DC). They are used as accelerators to boost the compute power of individual server nodes and to improve the overall power efficiency. Meanwhile, DC infrastructures are being redesigned to pack ever more compute capacity into the same volume and power envelopes. This redesign leads to the disaggregation of the server and its resources into a collection of standalone computing, memory, and storage modules.**

To embrace this evolution, we developed a platform that decouples the FPGA from the CPU of the server by connecting the FPGA directly to the DC network. This proposal turns the FPGA into a disaggregated standalone computing resource that can be deployed at large scale into emerging hyperscale data centers.

This paper describes an infrastructure which integrates 64 FPGAs (Kintex® UltraScale® XCKU060) from Xilinx® in a 19" × 2U[1] chassis, and provides a bi-sectional bandwidth of 640 Gb/s. The platform is designed for cost effectiveness and makes use of hot-water cooling for optimized energy efficiency. As a result, a DC rack can fit 16 platforms, for a total of 1024 FPGAs + 16 TB of DRR4 memory.

I. Introduction

Data-center disaggregation refers to the break-up of the traditional server architecture into a collection of standalone and modular computing, memory, and storage resources. In practice, it translates into the transmutation of the traditional rack and blade servers into sled and micro-servers. This move is purely driven by the performance-per-dollar metric, which is improved by increasing the density of the servers and by sharing resources, such as power supplies, PCB backplanes, cooling, fans, networking uplinks, and other management infrastructure.

The density of a server is increased when just about all extraneous parts to the processor, to the local memory, and to the boot storage are stripped from the motherboard. This pruning process leads to servers with shrunken form factors. For example, Facebook® has recently contributed the design of its system-on-chip (SoC) server called Mono Lake to the Open Compute Project. This single-socket server assembles an Intel® Xeon®-D, 32 GB of DRAM and 128 GB of boot storage on a motherboard of just 210×160 mm.

At the same time, FPGAs are getting their foot in the door of DCs: They start to be used for offloading and accelerating application-specific workloads, such as network encryption, web-page ranking, memory caching, deep learning, high-frequency trading, user authentication, and privacy protection. While graphics processing units (GPU) offer unprecedented compute density with 10s of TFlops for dual/single/half-precision floating-point and fixed-point operations, the advantages of FPGAs are their flexibility for building a custom control path and deep execution pipelines. At the level of DC applications, this translates into substantial improvements in energy efficiency, price per performance, and latency.

Unfortunately, these advantages cannot be realized using the common approach of deploying high-end FPGA boards as PCIe-attached extension cards in standard 2-socket servers, because the additional cost and power consumption diminish the energy-efficiency gains and cost savings. Moreover, this bus attachment limits the number of FPGAs that can be deployed per server and therefore hinders the potential of offloading large-scale applications. Finally, the form factor of the traditional PCIe interface is typically no longer compatible with the emerging dense and cost-optimized servers.

We observed those trends and concluded that if FPGAs want to continue gaining ground in future DCs, a change of paradigm is required for the FPGA-to-CPU attachment as well as for the form factor of the FPGA cards.

This paper showcases a hyperscale infrastructure based on the concepts of disaggregated FPGAs that we introduced in [1] and evaluated in [2][3].

II. Disaggregated FPGAs

In [1], we advocated the disaggregation of the FPGA from the server by means of an integrated 10GbE network controller interface (NIC) that connects the FPGA directly to the DC network as a standalone resource. This approach sets the FPGA free from the CPU and its traditional bus attachment, and becomes the key enabler for large-scale deployments of FPGAs in DCs.

Figure 1a shows the implementation of such a standalone disaggregated FPGA based on a Kintex® UltraScale™ XCKU060 from Xilinx. The card is physically similar to a half-length low-profile merchant PCIe x16 card without its two Small Form-factor Pluggable (SFP+) optical transceiver cages. Instead, the high-speed serial links are routed to the card-edge connector and operated over the backplane version of the 10 Gb Ethernet standard (10GBASE-KR). This configuration saves 30% of board space, 2/4 Watt of power consumption, and 50/100$ per 10 Gb/s duplex interconnect.

One major implication of the FPGA being dismantled from the server host is that the card must be turned into a self-contained appliance capable of executing tasks that were previously under the control of a host CPU. These tasks include the ability to perform power-up and -down actions, to hook itself up to the network after power-up, and to perform all sorts of local health-monitoring and system-management duties. On the disaggregated FPGA card, these tasks are handled by a pervasive 32-bit ARM® controller implemented

[1] 1U = one rack unit = 1.75 inches (44.45 mm)

* Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. This is the accepted version of the article published by IEEE: 10.1109/HOTI.2017.13

with a programmable system-on-chip (PSoC*) device from Cypress*.

III. Platform Sled

At the time of writing, DC networks are being upgraded to 40/100 GbE. Attaching each FPGA to a 100 GbE network is not justified and too expensive. Instead, the platform assembles a cluster of 32 FPGAs onto a passive carrier board, and interconnects them via an Ethernet switch. The switch is then considered as the network point-of-delivery and its fast up-links are used to expose the individual FPGAs to the DC network.

Figure 1b shows the passive carrier board with 32 connectors (organized into 4 banks of 8 connectors) for plugging 32 disaggregated FPGA cards, referred here as FPGA modules, each about the size of a double-height dual inline memory module (140×62 mm). Each FPGA module connects via a 10 GbE link to the south side of an Intel FM6000 Ethernet L2/L3/L4 switch, for a total of 320 Gb/s of aggregate bandwidth. The north side of the FM6000 switch connects to eight 40 GbE up-links, which expose the FPGA cluster to the DC network with another 320 Gb/s. This provides a uniform and balanced (no over-subscription) distribution between the north and south links of the Ethernet switch, which is desirable when building large and scalable fat-tree topologies (a.k.a. Folded Clos topologies). The Ethernet switch is visible in the center of Figure 1b. It provides the same aggregate throughput as a top-of-rack switch (i.e., 640 Gb/s) and was shrunk down to the size of a smart phone (140×62 mm) to fit vertically into a 2U-height chassis.

A fully populated carrier board is referred to as a sled. Its various I/O voltage rails are generated by two shared power controllers (cf. modules at the far left and far right of Figure 1b), and the entire sled is managed by a 64-bit T4240 service processor from Freescale*/Nxp*/Qualcomm* running Fedora* 23.

The sled implements a universal serial bus (USB) between every PSoC of the FPGA modules and the service processor. This dedicated management connection is used to transport a new bitstream to a PSoC when the reconfiguration of an FPGA is requested. Alternatively, a new partial bitstream can also be delivered over the DC network to an internal configuration access port (ICAP) of the FPGA.

IV. Platform Chassis

Two sleds fit a 19" × 2U chassis, for a total of 64 FPGA modules. Figure 2 shows the 10 GbE and 40 GbE interconnection networks between the various connectors of such an assembly. The chassis implements two identical sleds, S0 and S1, each consisting of the following interconnects: the red wiring within a sled corresponds to 10 GbE links connecting the 32 FPGA modules to the south side of the FM6000 Ethernet switch. The blue wiring within a sled corresponds to 40 GbE up-links connecting the north side of the same Ethernet switch to 8 Quad Small-Form-factor Pluggable (QSFP) transceivers. The purple wires correspond to 10 GbE links which provide a low-latency ring topology between every four neighboring FPGA modules of a given sled. The green wiring also consists of 10 GbE links that interconnect two sleds for providing a redundant path to failover from the Ethernet switch of one sled to the switch of the neighbor sled. Finally, the black wiring between pairs of neighboring slots provides a PCIe x8 Gen3 interface.

The FPGA platform achieves its high packaging density by implementing a module every 7.6 mm. This very small small stride does not allow air-cooled heatsinks and fans. Hence, we deployed a combination of a passive cooling solution at the FPGA module level and an actively cooled element at the chassis level. Our implementation is done by replacing the FPGA lid with a custom-made heat spreader (cf. Figure 3) that allows the transport of the thermal energy laterally from the chip away to the borders of the module board, where the heat spreader is then coupled to an active water-cooled heat sink (cf. blue rails in Figure 4). The passive heat sink is built using standard PCB lamination processes and materials.

V. Applications

In [2], we first compared the network performance of our disaggregated FPGA with that obtained from bare-metal servers, virtual machines, and containers. The results showed that standalone disaggregated FPGAs outperform them in terms of network latency and throughput by a factor of up to 35x and 73x, respectively. We also observed that the Ethernet NIC integrated within the FPGA fabric was consuming less than 10% of the total FPGA resources.

The first application that we ported was a distributed text-analytics application [3]. We compared it with i) a SW-only implementation and ii) an implementation accelerated with PCIe-attached FPGAs. The results showed that the disaggregated FGPAs outperformed the two other implementations, with improved latency, throughput, and, latency variation by a factor of 40, 18, and 5, respectively.

These results confirm our assumptions about the performance and efficiency of the platform. The first large-scale applications that we target include a cloud deployment, a deep-learning application (using reduced-precision logic), an HPC application (e.g., stencils), and a data-management application that combines FPGA modules with NVMe drive modules within the same platform.

VI. Related Work

The prevailing way of incorporating FPGAs in a server is by connecting them to the CPU through a PCIe interface and to use them as co-processors. The survey by Kachris et al. [4] shows that this practice remains the common case for the FPGA-based accelerators that have recently been proposed and implemented to off-load some of the most widely used cloud computing applications. The Microsoft* Catapult system, which pioneered the use of FPGAs at scale in DCs, was built in a similar way: a daughter card equipped with a single high-end FPGA was added to every Microsoft Open Cloud Server [5].

However, this PCIe attachment has two major issues in DC deployment. First, the power consumption of a server is an order of magnitude higher than that of an FPGA. Hence, the power efficiency that can be gained by offloading tasks from the server to 1 or 2 FPGAs is very limited [6]. Second, in DCs, the workloads are heterogeneous and run at different scales. Therefore, the scalability and the flexibility of the FPGA infrastructure are vital to meet the dynamic processing

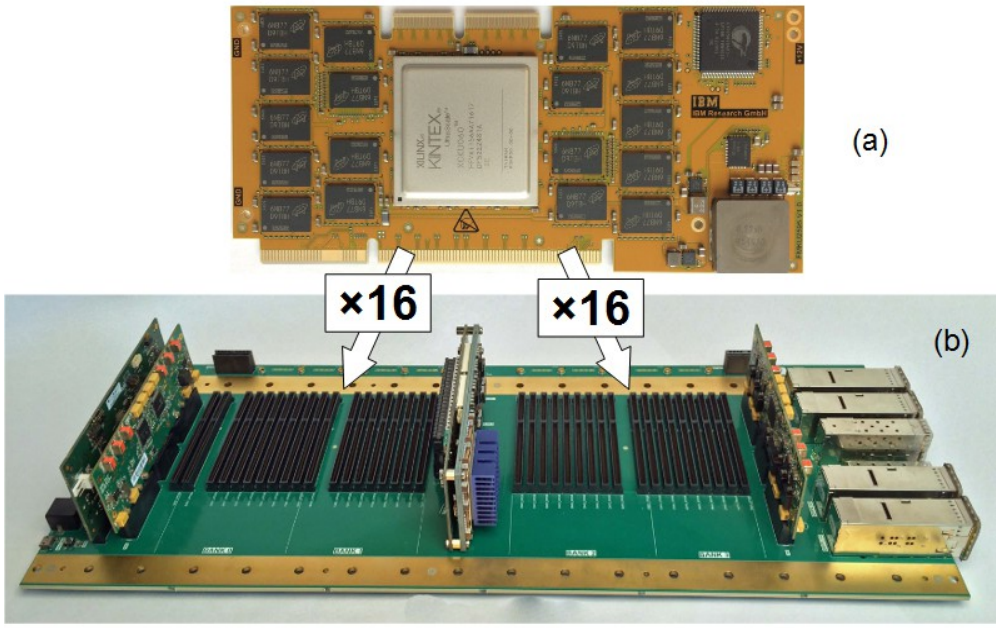

Figure 1: (a) The disaggregated FPGA and (b) the carrier board.

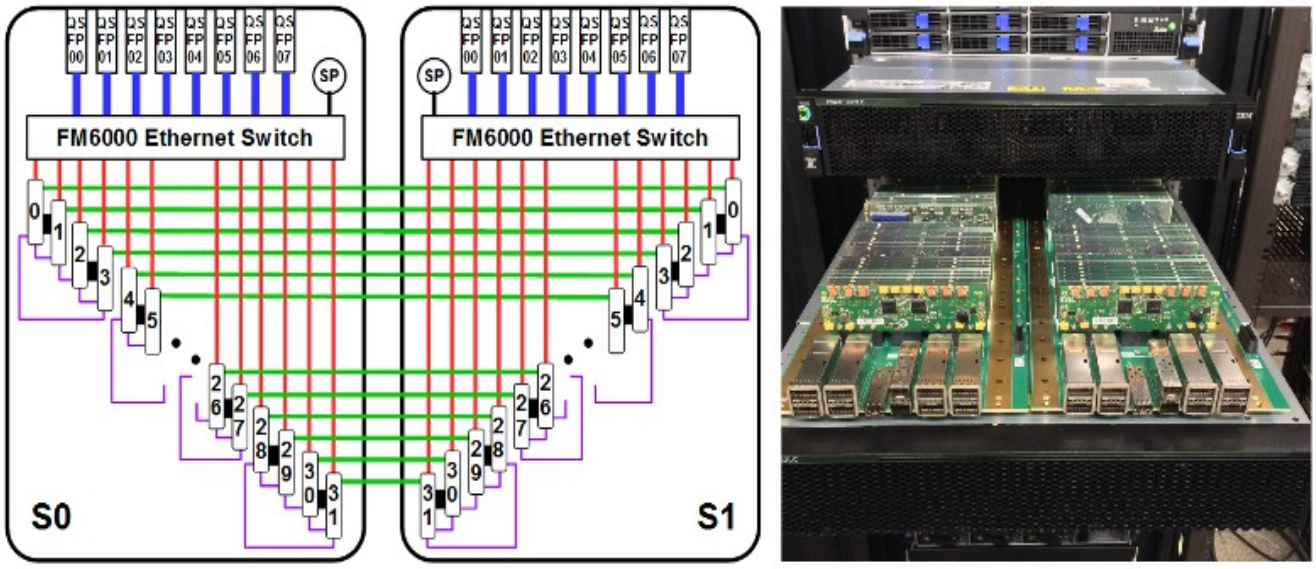

Figure 2: Block diagram of the PCB interconnect (left) and photo of two sleds in a chassis (right).

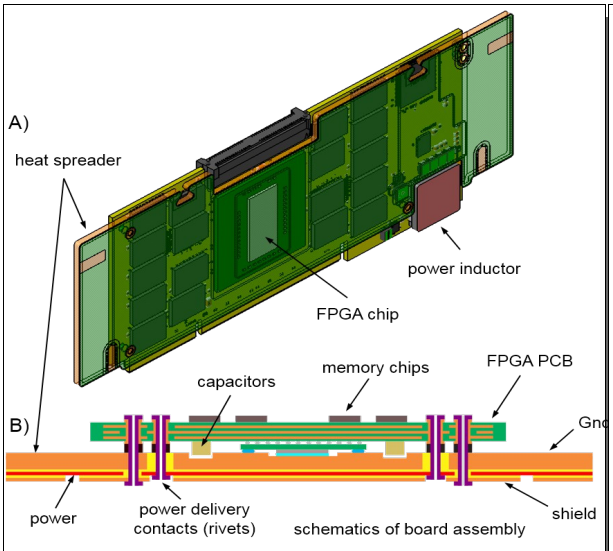

Figure 3: Passive cooling concept

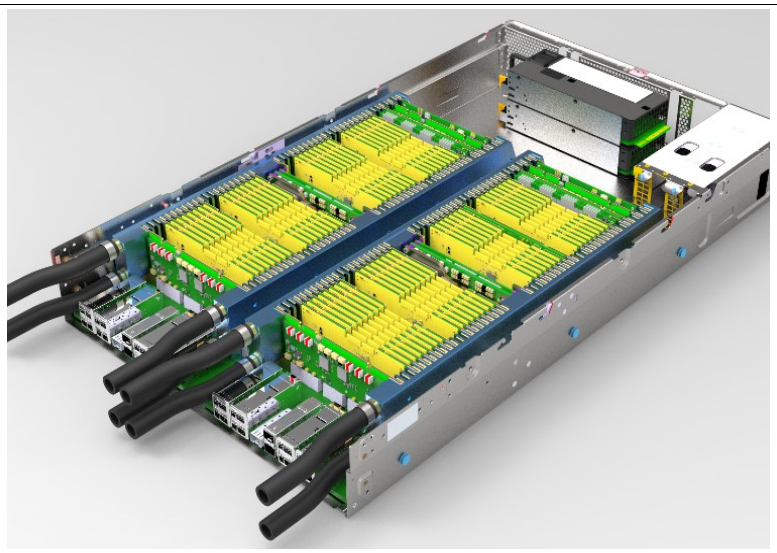

Figure 4: Rendering of the 2U × 19"chassis.

demands. With PCIe attachment, a large number of FPGAs cannot be assigned to run a workload independently of the number of CPUs. The above-mentioned Catapult system got around that limitation by implementing a secondary network between FPGAs, at the cost of increased complexity and loss of homogeneity in the DC. These drawbacks were later alleviated in Catapult v2 by placing the FPGA between the servers' NIC and the Ethernet network in a so-called "bump-in-the-wire" architecture [7]. This new arrangement enables FPGAs to reach each other over the DC network, but does not remove the dependency between the number of FPGAs and the number of servers.

Enabling the FPGAs to generate and consume their own networking packets independently of the servers opens new opportunities, such as linking multiple FPGAs with low latency and in any type of topology. For example, multiple FPGAs can be configured into a pipeline, a ring or a tree according to the application demands.

Finally, as FPGAs become plentiful in hyperscale data centers, cloud vendors are willing to offer them for rent to their users in a similar way as a standard server. The SuperVessel cloud from IBM* [8] and the EC2 F1 Instances from Amazon* [9] are two emerging ecosystems that propose remote access to FPGAs in the cloud for students, developers, and other customers. The first offer is limited to a single FPGA attached to a CPU via a PCIe bus or via a coherent accelerator processor interface (CAPI). The second provider can instantiate up to 8 PCIe-attached FPGAs per server.

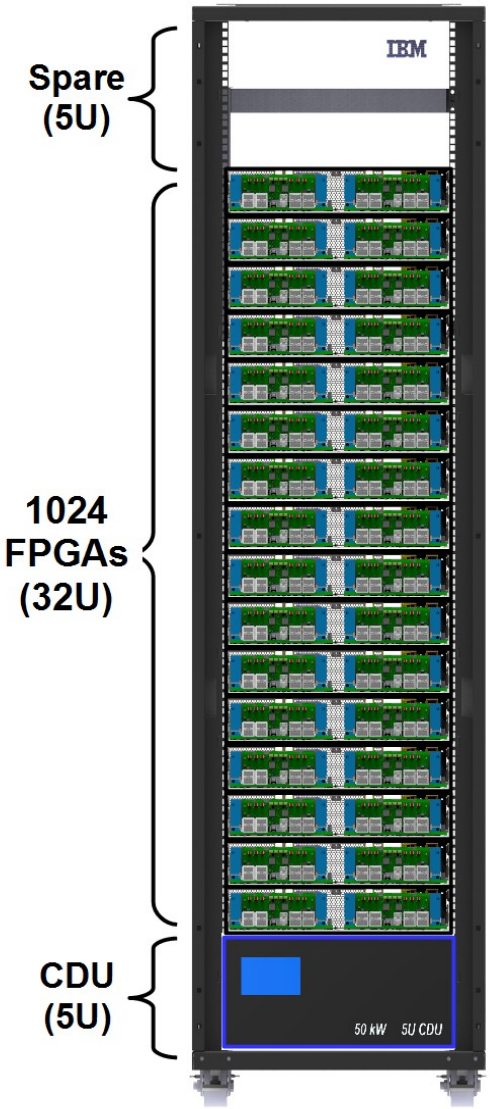

Figure 5: Rendering of a 42U rack equipped with 16 chassis.

VII. Summary

Our platform paves the way for the large-scale use of standalone disaggregated FPGAs in DCs. This deployment is particularly cost- and energy-efficient. First, the number of spread-out FPGAs becomes independent of the number of two-socket servers. Second, a large amount of network cables and transceivers has been removed and replaced by PCB traces inside a passive backplane. Finally, this network attachment promotes the FPGA to the rank of remote peer processor, which opens new perspectives for using them in a distributed fashion.

References

[1] J. Weerasinghe et al., "Enabling FPGAs in hyperscale data centers," in 2015 IEEE International Conference on Cloud and Big Data Computing (CBDCom), Beijing, China, 2015.

[2] J. Weerasinghe et al., "Disaggregated FPGAs: Network performance comparison against bare-metal servers, virtual machines and Linux containers," in IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, 2016.

[3] J. Weerasinghe et al., "Network-attached FPGAs for data center applications," in IEEE International Conference on Field-Programmable Technology (FPT '16), Xian, China, 2016.

[4] C. Kachris and D. Soudris, "A survey on reconfigurable accelerators for cloud computing," in 26th International Conference on Field Programmable Logic and Applications (FPL), Lausanne, Switzerland, 2016.

[5] A. Putnam et al., "A reconfigurable fabric for accelerating large-scale datacenter services," in Proceeding of the 41st Annual International Symposium on Computer Architecture," ser. ISCA'14. Piscataway, NJ, USA: IEEE Press, 2014.

[6] H. Giefers et al., "Analyzing the energy-efficiency of dense linear algebra kernels by power-profiling a hybrid CPU/FPG system," in IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors (ASAP), Zurich, Switzerland, 2014.

[7] A.M. Caulfield et al., "A cloud-scale acceleration architecture," In Proceedings of the 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Taipei, Taiwan, 2016.

[8] "SuperVessel cloud" [Online]. Available: https://www.ptopenlab.com/

[9] "Amazon EC2 F1 Instances," [Online]. Available: https://aws.amazon.com/ec2/instance-types/f1/

*→These are trademarks or registered trademarks of the respective companies in the United States and other countries. Other product or service names may be trademarks or service marks of IBM or other companies.

Acknowledgments

This work was conducted in the context of the joint ASTRON and IBM DOME project and was funded by the Netherlands Organization for Scientific Research (NWO), the Dutch Ministry of EL&L, and the Province of Drenthe, the Netherlands. We would like to thank Martin Schmatz, Ronald Luijten and Andreas Doering who initiated this new packaging concept for their microserver needs. Special thanks go to Ronald Otter, from Roneda PCB Design & Consultancy, for the design of the PCBs, and to Alex Raimondi, from Miromico AG, for the bring-up the FPGA boards.