Survey and Future Trends for FPGA Cloud Architectures

Hafsah Shahzad[*], Ahmed Sanaullah[**], and Martin Herbordt[*]

[*]CAAD Lab, ECE Department, Boston University
[**]Red Hat Inc.

2021 IEEE High Performance Extreme Computing Conference (HPEC) | 978-1-6654-2369-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/HPEC49654.2021.9622807

***Abstract*—In the last five years, FPGA presence in the cloud has gone from near zero (except for deeply embedded devices) to a large fraction of all high-end FPGAs sold. This is because FPGAs offer uniquely the performance, power, and flexibility needed to support the diversity and dynamicity of cloud workloads. We begin by observing that, although FPGAs are widespread, they cannot be randomly deployed as part of cloud infrastructure. Any FPGA cloud architecture must satisfy a number of constraints placed by the cloud provider. As a result, FPGA use in the cloud is non-uniformly distributed and motivated by the specific advantages and limitations that each unique architecture offers. In this survey, we provide an exploration and analysis of the trends in existing cloud FPGA architectures that highlight this complex relationship between architectures and system requirements. This allows us to identify novel architectures that are likely to offer substantial benefits for cloud workloads.**

I. Introduction

The demand for data center services is likely to continue growing exponentially [1]. As Moore's Law has slowed and the computational overhead and complexity of cloud workloads continue to rise and evolve, FPGAs offer a promising mechanism to address the paradox of simultaneously combining performance, power, and flexibility. They can be deployed almost anywhere in the data center in order to accelerate compute, storage, and network, as illustrated in Figure 1. Due to the immense benefits of FPGAs, their use in data centers is expected to grow at a Compound Annual Growth Rate (CAGR) of 48% between 2020 and 2027 [2].

While FPGAs offer a number of benefits in the data center, the choice of which particular architecture to leverage is complex and non-trivial; we simply cannot place any number and size of these devices anywhere in the cloud. The extreme need for cost-effectiveness leads to emphasis on size, power, cooling, compatibility, automation, and in-place upgradability, all while ensuring that specific needs of cloud workloads are met in terms of performance, memory, server capability, reliability, security, and network connectivity. For example, as illustrated in Figure 2, architectures can offer a different set of key benefits because of FPGA placement and connectivity. Thus, based on a system's requirements, certain FPGA cloud architectures can be substantially more advantageous than others.

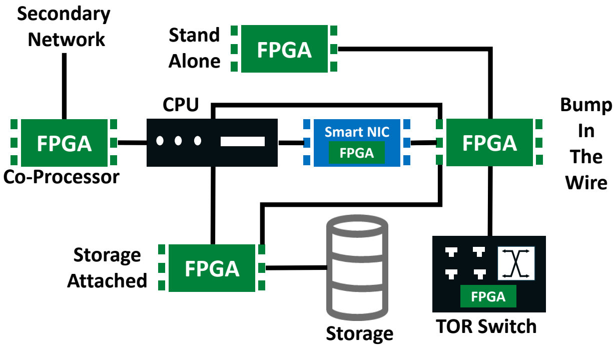


Fig. 1. Deployment versatility of FPGAs in the data center

Given the non-trivial relationship between system requirements and architectures, surveys of FPGA cloud architectures can play an immensely important role in helping deliver on the promise of FPGAs in the cloud. Specifically, these surveys can highlight the key advantages and limitations of individual architectures, which in turn make it easier to describe and compare systems without getting bogged down by low level implementation details. These surveys can also provide an 'innovation guide' to the vendors, including FPGA chip/board manufacturers and FPGA cloud system architects, about the features and limitations critical to their customers. Moreover, if a new system is to be brought online, these types of surveys can help reverse-engineer the best suited architecture based on a given set of requirements and constraints.

While there are several surveys that analyze different facets of cloud FPGAs [3]–[20], currently there is less work that addresses the architecture space. Relevant prior work is primarily discussions in support of specific technical contributions [21]–[25]. These are generally brief and based on broad categories and assumptions

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

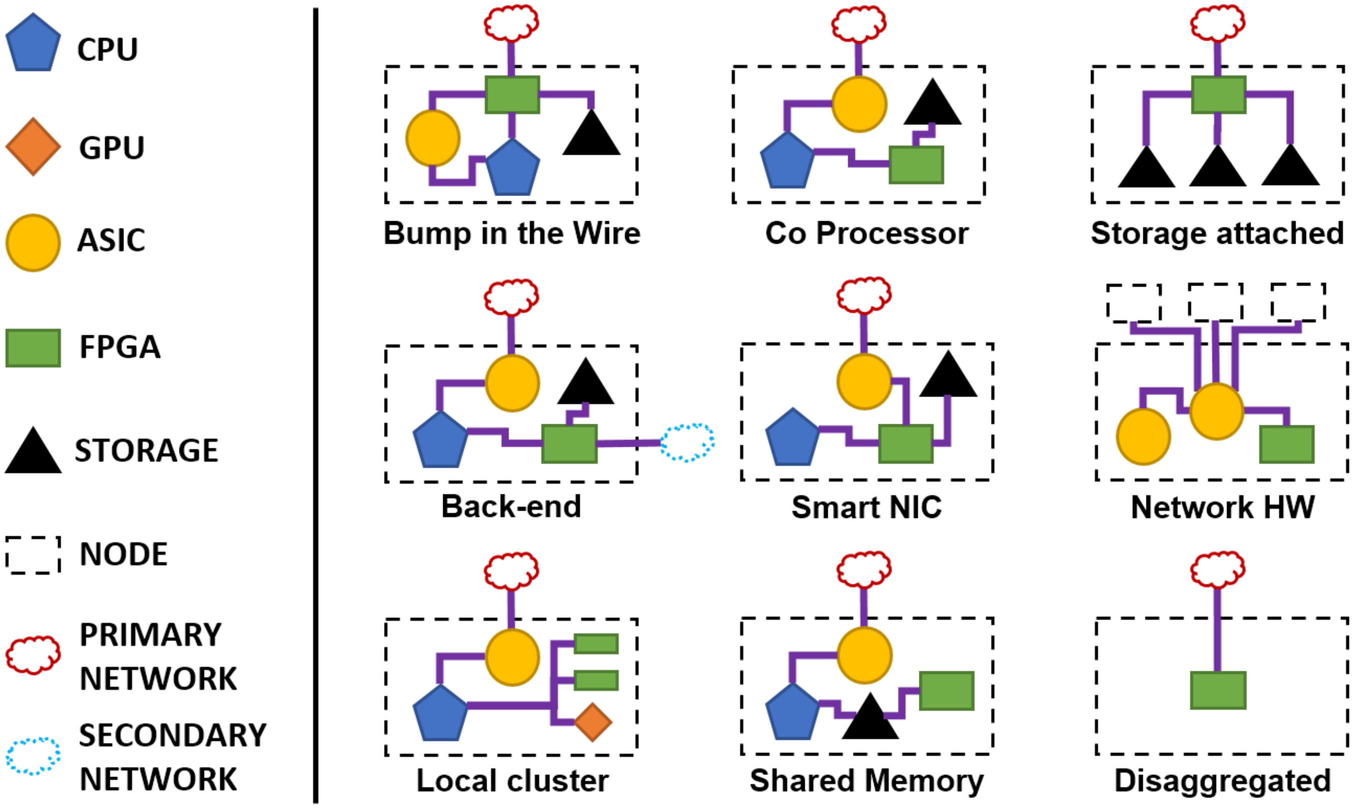


Fig. 2. Examples of common FPGA architectures. Potential benefits for each architecture include: i) **Bump-in-the-Wire:** Large scale compute, network and storage acceleration, ii) **Co Processor:** Local compute acceleration, iii) **Storage attached:** Local storage acceleration, iv) **Back-end cluster:** Ultra low latency, rack scale FPGA-FPGA communication, v) **Smart NIC:** Local network acceleration, vi) **Network HW:** Flexible routing/switching protocols, vii) **Local Cluster:** Multi-accelerator system, viii) **Shared Memory:** Cache coherent acceleration, and ix) **Disaggregated:** High infrastructure utilization.

that do not capture all of the unique set of advantages and challenges of different architectures. For example, the taxonomy in [21] captures FPGAs as in-node CPU compute accelerators, but does not extend to a number of other configurations, such as computational storage devices [26]–[28], stand-alone devices without a CPU in the node, or devices deployed outside of nodes, e.g. in TOR switches [29], [30].

The goal of this paper is to discuss cloud FPGA architectures with sufficient depth such that the relationship between design choices and constraints can be analyzed and trends can be identified, but while abstracting away low-level implementation details such as specific chips/boards and communication protocols. To do so, we survey existing cloud FPGA deployments in the context of the following central questions:

- **Type of FPGA board:** Are off-the-shelf boards used, or are the requirements specific or strict enough that a custom board is needed?
- **Placement of FPGAs in the system:** What type of components have FPGAs? Are they shared?
- **Network connectivity:** Do FPGAs have direct access to any inter-node networks?
- **Intra-node connectivity:** What significant devices can an FPGA talk to within a node? How?
- **Use cases:** Who is using the FPGAs (provider, user, etc.) and for what workloads?

Since the answer to the above can vary substantially based on the strictness of the constraints, we separate the analysis into production and research architectures. This allows us to both derive meaningful and valuable trends, as well as to plot a possible road map for cloud FPGA architectures, since research systems naturally point to future production systems. We also highlight promising, novel areas of innovation in the cloud FPGA architecture space that are currently not part of any production or research work, but may offer substantial value.

The specific contributions of this paper are:

- We survey existing cloud FPGA architectures deployed in both production and research systems.
- We classify these cloud FPGA architectures and use this classification to analyze important trends.
- We highlight several promising areas of future innovation in the cloud FPGA architecture space.

More generally we expect this work to serve different purposes to different communities, with the survey and

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

references providing an introduction to FPGAs-in-the-cloud to non-experts and the taxonomy and predictions being useful (or at least provocative) to practitioners.

The rest of this paper is organized as follows. Section II presents a taxonomy for classifying cloud FPGA architecture. Based on this taxonomy, Section III discusses existing production cloud FPGA architectures. Then, Section IV extends this discussion to cover existing research systems. Section V highlights potential future innovation derived from the taxonomy. Finally, Section VI gives the conclusion.

II. Taxonomy

The taxonomy is based on the critical questions highlighted in the previous section: **A) type of FPGA boards**, **B) placement of FPGAs in the system**, **C) network connectivity**, **D) intra-node connectivity**, and **E) use cases**.

A. Type of FPGA boards

Given that cloud providers are not currently creating their own FPGAs, the smallest unit of differentiation is the FPGA board; both **a) Off-the-shelf** and **b) Custom** are widely used. Economic advantages depend on scale of deployment and provider development infrastructure. Given the latter, custom boards still have higher start-up and upgrade costs, but may be cheaper in large quantities. But the advantage of scale also affects off-the-shelf economics as the provider has the market power to affect price and board features.

With custom FPGA boards just about any attribute can be varied, such as number/types/bandwidths of I/O ports, FPGA family, off-chip memory type and size, form factor, and other on-board devices. This ensures that the boards closely match the specifications/requirements of the target system, from computation to cooling. None-the-less, off-the-shelf boards are available for every (currently) sizable usage domain, including SoCs [31], [32], node-level networking [33]–[39], NoCs [40], data center switches [29], [30], [41], [42] and storage [26]–[28], [43], [44].

B. Placement of FPGAs

FPGAs can be placed in either a **a) distributed** or **b) centralized** manner. Having a distributed FPGA placement means that compute/storage nodes have their own FPGAs, and thus do not have to compete for the resource. This leads to more offload capability, greater reliability (since FPGA failure does not affect on other compute/storage nodes), and reduces security concerns since offloads for different nodes can be isolated. It is also possible to place FPGAs in a centralized manner, typically inside the networking nodes (e.g. in switches as ASIC-FPGA, CPU-FPGA, or FPGA only circuits). Substantially fewer FPGAs are needed for such a deployment; this typically translates to easier management, lower power consumption, lower TCO, smaller average node sizes, and potentially higher performance since expensive high-end FPGAs can be used (and upgraded more frequently).

C. Network connectivity

Within each node, it is possible for FPGAs to be **a) not connected to any network** or connected to **b) the primary network** and/or **c) a secondary network**. Being connected to the primary data center network enables FPGAs to intercept/accelerate network traffic to the node, as well as achieve data-center-wide scalability for FPGA workloads since multiple FPGAs can directly communicate with each other. However, the circuitry needed to support this FPGA position can consume a significant portion of FPGA resources. This includes circuits to support high resiliency since the FPGAs can be a single point-of-failure: an entire node can become unstable in the case of an FPGA error. In the case of secondary network connectivity, FPGAs can communicate across nodes with significantly more flexibility in the topology used (e.g. mesh, torus, switched), as well as the communication protocol, all of which can lead to ultra low latencies. However, using a custom network configuration means that complex router hardware, routing algorithms, and switch arbitration policies may need to be implemented on each FPGA. Moreover, complex cabling may be required, which can add a significant burden to the overall data center architecture [21].

D. Intra-node connectivity

FPGAs within a node can be **a) not connected to any other significant device** (i.e. be a *Disaggregated* resource) or connected to one or more devices: **b) CPUs**, e.g. through PCIe and possibly with cache coherence using interconnects such as CCIX [45], CXL [46], or CAPI [47]; **c) Other FPGAs**, e.g. through a PCIe switch and/or using direct and programmable interconnects; **d) GPUs**, e.g. through a PCIe switch; **e) ASICs**, e.g. through multiple potential forms of connectivity depending on the ASIC and nature of coupling such as NIC or tensor processor; **f) Storage devices** through the device-specific interface, e.g. SPI for flash and DDR for SDRAM.

E. Use cases

Use cases have a substantial impact on architecture, since cloud providers must ensure workload requirements are met (e.g. performance) without compromising on core

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

aspects (e.g. security, reliability). Common cloud FPGA usage domains are the following. **a) Customer applications:** Customers can develop, simulate, debug and compile their custom FPGA logic, as well as scale their infrastructure and change resources according to their workload demands. A wide pool of applications can be deployed, e.g. in genomics, financial analytics, computational fluid dynamics, video processing, transcoding, and security. Several development environments are available so users do not need to write their own HDL code [48]. **b) Provider Application as a Service (AaaS):** The cloud provider supports a limited set of customer applications by developing the FPGA design themselves: only the necessary APIs and high level design parameters are exposed. This model ensures high performance and resilience at the expense of reducing customer access to the entire FPGA. **c) Provider applications:** Cloud providers use FPGAs to accelerate their internal workloads, e.g. SDN, as well as save CPU resources that can then be rented to the customer.

III. Production Architectures

In this section we discuss production cloud FPGA systems that are in widespread or large-scale use.

A. Overview

Perhaps the most widely deployed production system is Microsoft's unique Catapult v2 [21], which has FPGAs in most Azure and Bing SKUs in a *Bump-in-the-Wire* configuration: FPGA sits between the TOR, NIC ASIC and CPU, hence enabling data-center-wide communication within tens of microseconds of latency. These hundreds of thousands of FPGAs (or more) are used for both internal (e.g. network packet processing [49], Bing search [16]) and external workloads (e.g. Machine Learning inference as a service [16]), with HPC workloads also found to be plausible [50].

Another type of a widespread production system is the single node accelerator model, which leverages FPGAs in either a *Coprocessor* configuration, or as a *Local Cluster* where devices are interconnected either via a PCIe switch or using direct FPGA-FPGA interconnects. A number of cloud providers such as AWS [51], Huawei [52], Baidu [53], Tencent [54], Nimbix [55] and Alibaba [56] use this model. These systems are used by customers to run a wide pool of cloud native applications such as genomics, financial analytics, data acquisition, computational fluid dynamics, video processing, image processing, transcoding, security, and AI workloads [57]–[62]. There are also examples of these FPGAs being used by providers for their own workloads. Baidu uses FPGAs to accelerate its cloud-based storage, SQL queries, data security, search engine, and AI workloads [59], [63]. FPGA-based AI chips–such as Baidu's Kunlun for AI, Alibaba's Ouroboros for speech recognition, and Alibaba's Hanguang 800 for inference operations–are deployed in their cloud data centers [64]. Alibaba has reported 75% savings in TCO by using FPGAs to oversee product images on its e-commerce site [65]. In 2018 it reported over $30 billion retail on its website in a single day (compared to $5 billion on all US online and in-store retail on black Friday 2017); this was possible with its data center FPGAs being used to accelerate transactions and provide recommendations to users [66].

There are also systems that are widely deployed, but where there is insufficient publicly available information for analysis. Amazon has announced AQUA (Advanced Query Accelerator) nodes for its Redshift data warehouse, available through the RA3.16XL and RA3.4XL instances. These nodes use FPGAs to accelerate dataset filtering and aggregation [67], [68]. Baidu uses Smart NICs to improve virtualisation and workload performance [69]. OVHCloud also uses Smart NICs, but for network packet processing to mitigate Distributed-Denial-of-service (DDoS) attacks in its cloud traffic [70], [71]. Scaleflux CSD2000 SSDs are deployed by over 40 data centers globally [72]. An example is the Alibaba cloud, which uses Scaleflux CSD20004 in place of traditional SSDs on their storage nodes to accelerate applications such as MySQL, Aerospike, Oracle, and PostGreSQL [73]. Samsung Smart SSDs [26] are deployed at the Nimbix cloud where they accelerate Apache Spark, running queries up to 6x faster when using software from Bigstream [74]. Eideticom's computational storage processor [27] has been implemented in Barreleye G2 servers on Rackspace [75].

B. Architecture Trends

Figure 3 classifies production cloud FPGA architectures into seven taxa (using the taxonomy in Section II). To effectively compare these architectures and highlight trends, we avoid illustrating the taxonomy as a single tree. Overall, there are four major trends: 1) Boards, 2) Placement, 3) The relationship between network connectivity and use cases, and 4) Intra-node connectivity.

1) Boards

Figure 3a shows that a majority of vendors have used custom boards in their deployments due to the benefits discussed earlier. For Microsoft in particular, this was necessary since requirements for placing FPGAs in special HPC SKUs "constrained power to 35W, the physical size to roughly a half-height half-length PCIe expansion card (80mm x 140 mm), and tolerance to an

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

a)

Type of Board	
Custom	Off-the-shelf
[A], [B], [C2], [F], [H]	[N], [T]

c)

Network / Use	Consumer	PaaS	Provider
Primary		[C2]	[C2]
Secondary			
None	[A], [B], [F], [H], [T]	[N]	[A], [B], [H]

b)

Placement	
Centralised	Distributed
	[A], [B], [C2], [F], [H], [N], [T]

d)

Intra-node Connectivity				
FPGA	CPU	GPU	ASIC	Storage
[A], [F], [H]	[A], [B], [C2], [F], [H], [N], [T]		[C2]	[A], [B], [C2], [F], [H], [N], [T]

Fig. 3. Classification for the following production cloud FPGA architectures: Alibaba [A], Baidu [B], Microsoft Catapult v2 [C2], Amazon AWS F1 [F], Huawei [H], Nimbix [N] and Tencent[T].

inlet air temperature of 70○C at 160 lfm airflow" [21]. Custom boards are not required, however: Nimbix and Tencent both use off-the-shelf.

2) Placement

Figure 3b shows that all these systems deploy FPGAs in a distributed fashion. This is because: i) FPGA resources are easier to orchestrate, ii) FPGAs can be offered as bare-metal resources, which simplifies the tooling needed, and iii) FPGA failure affects only local resources, as opposed to potentially millions of nodes.

3) Network Connectivity - Use cases

Figure 3c shows two important trends. First, none of the systems uses a secondary network. This is likely because of: i) the cost and complexity of wiring a second network for potentially millions of nodes and additional networking hardware, ii) the potentially limited scalability if direct FPGA-FPGA connectivity is supported, and iii) high chip resource usage for building routers and securing the system. The second important trend is the relationship between network connectivity and use cases. Specifically, due to security and reliability constraints, systems that allow customers to offload their own applications do not support any direct network connectivity. Rather, this connectivity is only available if workloads are either internal, or if the offering is an application where only a limited set of APIs are exposed to the customer.

4) Intra-node Connectivity

Figure 3d shows four major trends. First, in all of the systems FPGAs communicate with the CPU over the PCIe slot. This emphasises the role of the CPU as being the core computational resource, whereas the FPGA is a complexity offload engine managed by the CPU. Second, all FPGAs are connected to some form of off-chip storage, typically a DDR memory chip on the same board. Third, no production system currently offers instances with FPGA-GPU connectivity. To the best of our knowledge, none of the cloud providers has placed GPUs and FPGAs within the same node. In terms of FPGA-ASIC connectivity, only Microsoft supports this since the FPGA must transparently process packets for the traditional NIC. Fourth, a majority of systems support *local* FPGA clusters through PCIe switches or direct FPGA-FPGA interconnects. This allows for high speed connectivity among a small number FPGAs.

IV. Research Architectures

We discuss systems that are presently in research and development and represent the most technologically plausible candidates for widespread future deployment.

A. Overview

One of the most commonly used research architectures is the cluster of *Back-End* tightly coupled FPGAs that deploys a secondary network using direct and programmable interconnects to connect FPGAs across nodes. Microsoft's Catapult V1 was a back-end system that connected multiple nodes in 6x8 tori [76]. It was demonstrated using Microsoft's Bing workloads; it is not clear whether is was ever part of a production cloud. Although this approach can substantially reduce FPGA-FPGA latency, it is difficult to scale beyond a single rack due to wiring requirements; in the general case it also requires each FPGA to implement a router to support the communication. Currently no such example can be found operating in the production cloud. Other research examples include the 2D torus of 64 FPGAs on Maxwell [77], Novo-G# with a 3D torus interconnect among 64 FPGAs [78]–[80], the Noctua system at the Paderborn Center for Parallel Computing [81]–[83] with point to point connections among its 16 FPGA nodes, and the Albireo nodes of the Cygnus supercomputer system at University of Tsukuba [84] with a 2D 8x8 torus.

Another research area proposed in [85], [86] involves *Channel-over-Ethernet* (CoE), which is a back-end, inter-FPGA Ethernet communication network using the OpenCL kernel programming. Communication is also possible via the host CPU with Infiniband as a primary network. The results demonstrate the feasibility of such a configuration as the system achieves a latency of 0.99µs for inter-FPGA communication via the secondary Ethernet switch as compared to 29.03µs via the host CPU. A drawback is that data are sent as packets so there is additional overhead, such as IP addresses and flags, that reduce the effective data rate [87].

Other research architectures include systems that support a *Local Cluster*, but where the communication scaling via direct interconnects is limited to a single

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

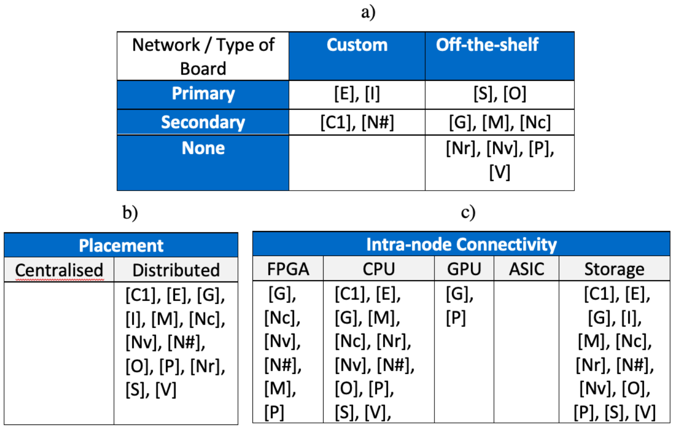
node. Examples include Novo-G (a former version of Novo-G#) that allowed eight FPGAs wired on the same node communicate directly [88]. Another example is the research systems currently deployed at the IBM SuperVessel Cloud [89] and the IBM Power8+CAPI cluster at the University of Texas, Austin [90] that use a *Shared Memory* cache coherency model.

A different approach is to directly connect FPGAs to the data center network as a standalone resource. Each FPGA can be accessed by a CPU or another FPGA resulting in good scalability. CloudFPGA at the IBM Zurich Research Lab has demonstrated that network-attached disaggregated FPGAs improve network latency and throughput over other configurations, e.g. SW-only, PCIe attached FPGAs, bare metal servers, and virtual machines [91], [92]. The authors have built data center rack scale prototype with 1024 FPGAs [93]. A drawback of such an architecture may be that FPGA-CPU communication is necessarily among separate nodes and has high latency. Another consideration is the increase in the number of TOR connections.

The Open Cloud FPGA Testbed (OCT) is another research system that connects off-the-shelf FPGA boards to the network and also to a host CPU via PCIe [94]. The testbed provides flexibility for cloud researchers to experiment with bare metal nodes, FPGAs' programming, and with FPGAs connected directly to the network and to one another [95]. University of Toronto SAVI testbed connects FPGAs to the primary network [96]. The authors in [3], [97] have demonstrated that virtualising FPGA resources on the SAVI testbed enables multiple regions within an FPGA device to support different designs using APIs such as OpenStack. Enzian [98] at ETH Zurich employs an FPGA as a node connected to the network on one end and coherently attached to a large server-class SoC on another node. Unlike Microsoft's *Bump-in-the-wire*, this system allows CPUs to either connect directly to the network or via the FPGA. Unlike other cache coherent systems, it allows the FPGA side of the cache coherency protocol to be extended and tailored [99].

B. Architecture Trends

Figure 4 classifies the research cloud FPGA architectures using the taxonomy defined in Section II.As we can see, with the exception of a few possibilities, research systems have explored different varieties of cloud architecture options. While production systems are bounded by several factors such as total-cost-of-ownership (TCO), power-usage-effectiveness (PUE), performance, resilience, modularity, scalability and security; research systems tend to enjoy greater degrees of freedom.



a)

Network / Type of Board	Custom	Off-the-shelf
Primary	[E], [I]	[S], [O]
Secondary	[C1], [N#]	[G], [M], [Nc]
None		[Nr], [Nv], [P], [V]

b)

Placement	
Centralised	Distributed
	[C1], [E], [G], [I], [M], [Nc], [Nv], [N#], [O], [P], [Nr], [S], [V]

c)

Intra-node Connectivity				
FPGA	CPU	GPU	ASIC	Storage
[G], [Nc], [Nv], [N#], [M], [P]	[C1], [E], [G], [M], [Nc], [Nr], [Nv], [N#], [O], [P], [S], [V],	[G], [P]		[C1], [E], [G], [I], [M], [Nc], [Nr], [N#], [Nv], [O], [P], [S], [V]

Fig. 4. Classification for selected research cloud FPGA architectures: Microsoft Catapult v1 [C1], Enzian [E], Cygnus[G], IBM cloudFPGA [I], Maxwell [M], Noctua [Nc], NARC [Nr] [100], Novo-G [Nv], Novo-G# [N#], Open Cloud Testbed [O], Power8+CAPI TACC [P], SAVI [S], IBM SuperVessel [V].

1) Boards

Figure 4a shows that custom boards are preferred if the proposed systems are *Disaggregated*, network attached (e.g Enzian and IBM CloudFPGA). Also, for earlier *Back-end* systems like Catapult v1 and Novo-G# a customised board allowed the system to increase transceiver count. However, we can see that recent *Back-end* and *Local Cluster* systems mostly use off-the-shelf boards. Systems with no inter-node communication network almost always use off-the-shelf boards.

2) Placement

To the best of our knowledge, no research systems are deployed in a centralized manner (Figure 4b).

3) Network Connectivity

Figure 4a shows that research systems are distributed evenly across the different network connectivity options. We also note that newer systems almost always have network connectivity, either primary or secondary. This helps scale the application across multiple FPGAs and achieve lower latency.

4) Intra-node Connectivity

Figure 4c shows three major trends. First, none of the research architectures connects an ASIC with an FPGA on the same node. Second, all systems have some form of off-chip storage. Third, GPUs are being employed on the same node as FPGAs, especially for highly parallel, SIMD-like workloads and communicate over a PCIe switch [84].

V. Potential Future Innovation

We identify areas of potential novelty that can be derived by traversing the categories in the taxonomy, and by comparing different sub-categories with what is already present in Figures 3 and 4.

Type of Boards: Potential novelty here is with modular boards that lie at the intersection of custom and commodity. Similar to what is commonly done with micro-controllers, semi-custom boards can be built by buying and connecting together off-the-shelf modules for different FPGA chips, memory chips, and interfaces (QSFP+, PCIe etc). This would allow providers to tailor boards to their specific requirements, reduce the penalties of designing a custom board (development costs, upgrade costs, probability of failure, time to market), and easily replace specific modules as needed (due to hardware failure or for regular upgrades).

Placement of FPGAs: While FPGAs have been used in high end network switches [41], [101], [102], their role is typically limited to providing the performance and flexibility needed to support changing protocols. However, there is currently no system that leverages TOR switches where FPGAs are responsible for implementing the entire switch hardware [29], [30].

Supporting such an architecture has a number of benefits. i) Customer offloads: Customers could use these TOR FPGAs to compute in the network e.g. for doing collective operations such MPI All-Reduce and Broadcast. ii) Provider offloads: Providers could leverage these FPGAs to implement services such as metering, accounting, analytics, and packet filtering. iii) Flexible networking: By combining FPGA based TORs with Bump-in-the-Wire FPGAs, a data-center-wide network could be created that does not rely on a standard protocol for communication. As a result, the communication latency could be reduced substantially. Alternatively, it may be possible to dynamically switch between different standard protocols based on the target workload.

Network Connectivity: A potential novelty here would be to support both Primary and Secondary network connectivity, either within the same FPGA, or through multiple tightly coupled FPGAs within the same node. This would effectively combine key benefits of Microsoft's Catapult v1 and v2, i.e. having ultra low latency for rack scale communications through custom interconnects and still supporting data-center scale FPGA-FPGA connectivity.

Intra-node connectivity and Use cases: The connectivity between FPGAs and CPUs is typically done using the PCIe bus. This is because existing use cases define the role of the FPGA as an offload engine for the CPU. However, a potential novelty here is supporting sufficient low-level electrical coupling, such as the FPGA has read-modify-write access to the CPUs Baseboard Management Controller and firmware. This would effectively turn the FPGA into a management and security controller for the CPU, and enable new *system administrator* use cases such as CPU firmware attestation.

VI. Conclusion

We present a survey of cloud FPGA architectures that explores the complex and non-trivial relationship between system requirements and deployment configurations and identifies areas of potential future innovation in this space. To help organize the survey, we use a taxonomy that abstracts away low-level implementation details while still highlighting advantages and limitations of a given architecture. Using this taxonomy, we classify both production and research systems; this in turn is used to demonstrate the major trends in cloud FPGA architecture. Finally, based on the findings of this survey, we identify several potential areas of innovation that are currently not explored in either production or research.

Acknowledgements

This work was supported, in part, by the NSF through award CCF-1919130; the NIH through award R44GM128533; by a grant from Red Hat; and by Xilinx and by Intel both through donated FPGAs, tools, and IP.

References

[1] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.

[2] Global Industry Analysts, Inc, "Data Center FPGA Market," https://www.researchandmarkets.com/reports/4804594/data-center-accelerators-global-market\#pos-0 [Last accessed: April 29, 2021].

[3] N. Tarafdar, T. Lin, D. Ly-Ma, D. Rozhko, A. Leon-Garcia, and P. Chow, "Building the Infrastructure for Deploying FPGAs in the Cloud," in *Hardware Accelerators in Data Centers*. Springer, 2019, pp. 9–33.

[4] A. Vaishnav, K. D. Pham, and D. Koch, "A Survey on FPGA Virtualization," in *FPL*, 2018, pp. 131–1317.

[5] K. Vipin and S. A. Fahmy, "FPGA Dynamic and Partial Reconfiguration: A survey of Architectures, Methods, and Applications," *Comp Surv*, vol. 51, no. 4, pp. 1–39, 2018.

[6] Knodel, Oliver and Genssler, Paul R and Spallek, Rainer G, "Virtualizing Reconfigurable Hardware to Provide Scalability in Cloud Architectures," in *Int. Conf. Adv Circuits, Electronics, and Micro-electronics*, 2017.

[7] Y. Zha and J. Li, "Virtualizing FPGAs in the Cloud," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020, pp. 845–858.

[8] D. Korolija, T. Roscoe, and G. Alonso, "Do OS abstractions make sense on FPGAs?" in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2020, pp. 991–1010.

[9] C. H. Yu, P. Wei, M. Grossman, P. Zhang, V. Sarker, and J. Cong, "S2FA: An Accelerator Automation Framework for Heterogeneous Computing in Datacenters," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. IEEE, 2018, pp. 1–6.

[10] R. A. Cooke and S. A. Fahmy, "Characterizing Latency Overheads in the Deployment of FPGA Accelerators," in *2020*

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

30th International Conference on Field-Programmable Logic and Applications (FPL). IEEE, 2020, pp. 347–352.

[11] J. Zhang, Y. Xiong, N. Xu, R. Shu, B. Li, P. Cheng, G. Chen, and T. Moscibroda, "The Feniks FPGA Operating System for Cloud Computing," in *Proceedings of the 8th Asia-Pacific Workshop on Systems*, 2017, pp. 1–7.

[12] C. Kachris and D. Soudris, "A Survey on Reconfigurable Accelerators for Cloud Computing," in *26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1–10.

[13] N. Mohammedali and M. O. Agyeman, "A study of Reconfigurable Accelerators for Cloud Computing," in *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, 2018, pp. 1–5.

[14] Blaiech, Ahmed Ghazi and Khalifa, Khaled Ben and Valderrama, Carlos and Fernandes, Marcelo AC and Bedoui, Mohamed Hedi, "A Survey and Taxonomy of FPGA-based Deep Learning accelerators," *Journal of Systems Architecture*, vol. 98, pp. 331–345, 2019.

[15] Yu, Xiaoyu and Wang, Yuwei and Miao, Jie and Wu, Ephrem and Zhang, Heng and Meng, Yu and Zhang, Bo and Min, Biao and Chen, Dewei and Gao, Jianlin, "A Data-Center FPGA Acceleration Platform for Convolutional Neural Networks," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 151–158.

[16] Chung, Eric and Fowers, Jeremy and Ovtcharov, Kalin and Papamichael, Michael and Caulfield, Adrian and Massengill, Todd and Liu, Ming and Lo, Daniel and Alkalay, Shlomi and Haselman, Michael and others, "Serving DNNs in Real Time at Datacenter Scale with Project Brainwave," *IEEE Micro*, vol. 38, no. 2, pp. 8–20, 2018.

[17] Matas, Kaspar and La, Tuan and Grunchevski, Nikola and Pham, Khoa and Koch, Dirk, "Invited Tutorial: FPGA Hardware Security for Datacenters and Beyond," in *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2020, pp. 11–20.

[18] S. Trimberger and S. McNeil, "Security of FPGAs in Data Centers," in *2017 IEEE 2nd International Verification and Security Workshop (IVSW)*. IEEE, 2017, pp. 117–122.

[19] S. Tian, W. Xiong, I. Giechaskiel, K. Rasmussen, and J. Szefer, "Fingerprinting Cloud FPGA Infrastructures," in *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2020, pp. 58–64.

[20] Wang, Xiuxiu and Niu, Yipei and Liu, Fangming and Xu, Zichen, "When FPGA Meets Cloud: A First Look at Performance," *IEEE Transactions on Cloud Computing*, 2020.

[21] A. M. Caulfield, E. S. Chung, A. Putnam, H. Angepat, J. Fowers, M. Haselman, S. Heil, M. Humphrey, P. Kaur, J.-Y. Kim *et al.*, "A Cloud-Scale Acceleration Architecture," in *49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2016, pp. 1–13.

[22] R. Skhiri, V. Fresse, J. P. Jamont, B. Suffran, and J. Malek, "From FPGA to support cloud to cloud of FPGA: State of the art," *International Journal of Reconfigurable Computing*, vol. 2019, 2019.

[23] Zhu, Zhuangdi and Liu, Alex X and Zhang, Fan and Chen, Fei, "FPGA Resource Pooling in Cloud Computing," *IEEE Transactions on Cloud Computing*, 2018.

[24] J. Weerasinghe, "Standalone disaggregated reconfigurable computing platforms in cloud data centers," Ph.D. dissertation, Technical University Munich, 2017.

[25] J. Weerasinghe, F. Abel, C. Hagleitner, and A. Herkersdorf, "Enabling FPGAs in Hyperscale Data Centers," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, 2015, pp. 1078–1086.

[26] Xilinx, "SmartSSD Computational Storage Drive," https://www.xilinx.com/applications/data-center/computational-storage/smartssd.html [Last accessed: April 29, 2021].

[27] Eideticomm, "NVMe Computational Storage," https://www.eideticom.com/ [Last accessed: April 29, 2021].

[28] Scaleflux, "Computational Storage," https://www.scaleflux.com/ [Last accessed: April 29, 2021].

[29] NewWaveDV, "32-Port Programmable Switch," http://newwavedv.com/wordpress/wp-content/uploads/2019/04/32-Port-Programmable-Switch-Datasheet.pdf [Last accessed: April 29, 2021].

[30] Huawei, "CloudEngine Is the Foundation of the Intent-driven Network," https://actfornet.com/ueditor/php/upload/file/20191206 /1575567949680852.pdf [Last accessed: April 29, 2021].

[31] Xilinx, "SoCs with Hardware and Software Programmability," https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html [Last accessed: April 29, 2021].

[32] Intel, "SoC FPGAs," https://www.intel.com/content/www/us/en/ products/programmable/soc.html [Last accessed: April 29, 2021].

[33] Xilinx, "Alveo SmartNIC Accelerator Card," https://www.xilinx.com/products/boards-and-kits/alveo.html [Last accessed: April 29, 2021].

[34] Xilinx, "Alveo SN1000 Accelerator Card," https://www.xilinx.com/applications/data-center/network-acceleration/alveo-sn1000.html [Last accessed: April 29, 2021].

[35] Inventec, "FPGA SmartNIC C5020X," https://ebg.inventec.com/en/product/Accessories/Smart\%20NIC\%20Card/Inventec\%20FPGA\%20SmartNIC\%20C5020X [Last accessed: April 29, 2021].

[36] Silicom, "FPGA SmartNIC N5010 Series," https://www.silicom-usa.com/pr/fpga-based-cards/fpga-intel-based/fpga-intel-stratix-based/silicom-fpga-smartnic-n5010_series/ [Last accessed: April 29, 2021].

[37] Napatech, "FPGA acceleration cards," https://www.napatech.com/products/ [Last accessed: April 29, 2021].

[38] Intel, "FPGA Programmable Acceleration Card D5005," https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/intel-fpga-pac-d5005/overview.html [Last accessed: April 29, 2021].

[39] Mellanox, "Innova-2 Flex Open Programmable SmartNIC," https://www.mellanox.com/files/doc-2020/pb-innova-2-flex.pdf [Last accessed: April 29, 2021].

[40] Acronix, "Speedster 7t FPGAs," https://www.achronix.com/product/speedster7t-fpgas [Last accessed: April 29, 2021].

[41] Cisco, "Nexus 3000 Switch Architecture," https://www.ciscolive.com/c/dam/r/ciscolive/us/docs/2018/pdf/BRKDCN-3734.pdf [Last accessed: April 29, 2021].

[42] Sdxcentral, "Broadcom Sharpens Tomahawk Switch Chips, Versatile SmartToR," https://www.sdxcentral.com/articles/news/broadcom-sharpens-tomahawk-switch-chips-versatile-smarttor/2020/12/ [Last accessed: April 29, 2021].

[43] Bittware, "Computational Storage," https://www.bittware.com/fpga/storage/ [Last accessed: April 29, 2021].

[44] Xilinx, "FPGAs: The Key to Accelerating High-Speed Storage Systems," https://www.flashmemorysummit.com/Proceedings2019/08-07-Wednesday/20190807_Keynote11\ _Xilinx_Raje.pdf [Last accessed: April 29, 2021].

[45] Synopsys, "An Introduction to CCIX," https://www.synopsys.com/designware-ip/technical-bulletin/introduction-ccix-2017q3.html [Last accessed: April 29, 2021].

[46] CXL, "CPU-to-Device Interconnect," https://www.computeexpresslink.org/about-cxl [Last accessed: April 29, 2021].

[47] OpenCAPI, "A New Standard for High Performance Memory, Acceleration and Networks," https://opencapi.org/2017/04/opencapi-new-standard-high-performance-memory-acceleration-networks/ [Last accessed: April 29, 2021].

[48] AWS, "Marketplace," https://aws.amazon.com/marketplace/search/results?x=0&y=0&searchTerms=fpga [Last accessed: April 29, 2021].

[49] Microsoft Azure, "Deploy ML models to FPGAs with Azure Machine Learning," https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/ [Last accessed: April 29, 2021].

[50] J. Sheng, C. Yang, A. Caulfield, M. Papamichael, and M. Herbordt, "HPC on FPGA Clouds: 3D FFTs and Implications for Molecular Dynamics," in *27th International Conference on Field Programmable Logic and Applications (FPL)*, 2017.

[51] AWS, "EC2 F1 Instances," https://aws.amazon.com/ec2/instance-types/f1/ [Last accessed: April 29, 2021].

[52] Huawei, "FPGA Cloud," https://www.huaweicloud.com/en-us/product/fcs.html [Last accessed: April 29, 2021].

[53] Baidu, "FPGA Cloud," https://cloud.baidu.com/product/fpga.html [Last accessed: April 29, 2021].

[54] Tencent, "FPGA Cloud Server," https://cloud.tencent.com/product/fpga [Last accessed: April 29, 2021].

[55] Nimbix, "FPGA Cloud," https://www.nimbix.net/what-is-an-fpga [Last accessed: April 29, 2021].

[56] Alibaba, "Alibaba FPGA Cloud," https://www.alibabacloud.com /help/doc-detail/108504.htm [Last accessed: April 29, 2021].

[57] AWS, "Accelerate applications using Amazon EC2 F1 FPGA instances," https://d1.awsstatic.com/events/reinvent/2019/Accelerate_applications_using_Amazon_EC2_F1_FPGA_instances_CMP314.pdf [Last accessed: April 29, 2021].

[58] TIBCO, "Customers See Success with TIBCO on AWS," https://www.tibco.com/blog/2018/11/26/tibco-customers-see-success-with-tibco-on-aws/ [Last accessed: April 29, 2021].

[59] Baidu, "Baidu ABC Platform," https://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/Baidu_ABC_Platform.pdf [Last accessed: April 29, 2021].

[60] Baidu, "AI Cloud," https://intl.cloud.baidu.com/product/abc-stack.html [Last accessed: April 29, 2021].

[61] Huawei, "FPGA as a Service in the Cloud," https://indico.cern.ch/event/669648/contributions/2838181/attachments/1581893/2500031/Huawei_Cloud_FPGA_as_a_Service_CERN_openlab.pdf [Last accessed: April 29, 2021].

[62] Intel, "Intel FPGAs Power Acceleration-as-a-Service for Alibaba Cloud," https://newsroom.intel.com/ news/intel-fpgas-power-acceleration-as-a-service-alibaba-cloud/#gs.uijjhu [Last accessed: April 29, 2021].

[63] TheNextPlatform, "Baidu Takes FPGA Approach to Accelerating SQL at Scale," https://www.nextplatform.com/2016/08/24/baidu-takes-fpga-approach-accelerating-big-sql/ [Last accessed: April 29, 2021].

[64] D. Ernst, "Competing in Artificial Intelligence Chips: China's Challenge amid Technology War," *Centre for International Governance Innovation, Special Report*, 2020.

[65] Xilinx Case Study, "Xilinx Powers Alibaba Cloud FaaS with AI Acceleration Solution for E-Commerce Business," https://www.xilinx.com/publications/powered-by-xilinx/xilinx-alibaba-case-study.pdf [Last accessed: April 29, 2021].

[66] Forbes, "Xilinx FPGAs: The Chip Behind Alibaba's Singles Day," https://www.forbes.com/sites/moorinsights/2018/11/29/xilinx-fpgas-the-chip-behind-alibabas-singles-day/?sh=5f2294e27e3b [Last accessed: April 29, 2021].

[67] AWS, "AQUA (Advanced Query Accelerator) for Amazon Redshift," https://pages.awscloud.com/AQUA_Preview.html [Last accessed: April 29, 2021].

[68] Blocks&Files, "A brief look at AWS Redshift's AQUA acceleration hardware," https://blocksandfiles.com/2019/12/05/amazon-aqua-data-warehouse-acceleration-hardware/ [Last accessed: April 29, 2021].

[69] Intel, "Intelligent Infrastructure Transformation," https://newsroom.intel.com/news/intel-baidu-drive-intelligent-infrastructure-transformation [Last accessed: April 29, 2021].

[70] Bittware, "How OVHcloud Uses FPGAs to Mitigate DDoS Attacks," https://www.bittware.com/resources/case-study-ovh/ [Last accessed: April 29, 2021].

[71] OVH, "Acceleration-as-a-Service leveraging Intel PAC," https://www.ovh.com/world/news/press/cp2541.ovh_launches_acceleration-as-a-service_leveraging_the_new_intel_programmable_acceleration_card_and_app_store_from_fpga_acceleration_partner_accelize [Last accessed: April 29, 2021].

[72] Blocks&Files, "Our Computational Storage Drives are Bigger, Faster and Cheaper than Ordinary SSDs," https://blocksandfiles.com/2021/02/22/scaleflux-ceo-hao-zhong-interview/ [Last accessed: April 29, 2021].

[73] Scaleflux, "Applications," https://www.scaleflux.com/news.html [Last accessed: April 29, 2021].

[74] TheNextPlatform, "Computational Storage Winds its Way Towards the Mainstream," https://www.nextplatform.com/2020/02/25/computational-storage-winds-its-way-towards-the-mainstream/ [Last accessed: April 29, 2021].

[75] PRNewsWire, "NVMe Production Ready System," https://www.prnewswire.com/news-releases/eideticom-ibm-rackspace-and-xilinx-demonstrate-worlds-first-pcie-gen4-nvm-express-production-ready-system-676532203.html [Last accessed: April 29, 2021].

[76] A. Putnam, A. M. Caulfield, E. S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. P. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Y. Xiao, and D. Burger, "A Reconfigurable Fabric for Accelerating Large-Scale Data Center Services," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, 2014, pp. 13–24.

[77] R. Baxter, S. Booth, M. Bull, G. Cawood, J. Perry, M. Parsons, A. Simpson, A. Trew, A. McCormick, G. Smart, R. Smart, A. Cantle, R. Chamberlain, and G. Genest, "Maxwell - a 64 FPGA Supercomputer," in *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, 2007, pp. 287–294.

[78] J. Sheng, C. Yang, and M. Herbordt, "Towards Low-Latency Communication on FPGA Clusters with 3D FFT Case Study," in *Proc. of the 6th Int. Symp. on Highly Efficient Accelerators and Reconfigurable Technologies*, 2015.

[79] A. D. George, M. C. Herbordt, H. Lam, A. G. Lawande, J. Sheng, and C. Yang, "Novo-G#: Large-Scale Reconfigurable Computing with Direct and Programmable Interconnects," in *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, 2016, pp. 1–7.

[80] J. Sheng, C. Yang, and M. Herbordt, "High Performance Dynamic Communication on Reconfigurable Clusters," in *The 26th IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2018.

[81] C. Plessl, "Bringing FPGAs to HPC Production Systems and Codes," in *H2RC'18 workshop at Supercomputing (SC'18)*, 2018, doi: 10.13140/RG.2.2.34327.42407.

978-1-6654-2369-4/21/$31.00 ©2021 IEEE

[82] Paderborn Center for Parallel Computing, University of Padeborn, “Noctua,” https://pc2.uni-paderborn.de/hpc-services/available-systems/noctua [Last accessed: April 29, 2021].

[83] TheNextPlatform, “Another Step towards FPGAs in Supercomputing,” https://www.nextplatform.com/ 2018/04/04/another-step-toward-fpgas-in-supercomputing/ [Last accessed: April 29, 2021].

[84] “Cygnus,” https://www.ccs.tsukuba.ac.jp/wp-content/uploads/sites/14/2018/12/About-Cygnus.pdf [Last accessed: April 29, 2021].

[85] R. Kobayashi, Y. Oobata, N. Fujita, Y. Yamaguchi, and T. Boku, “OpenCL-Ready High Speed FPGA Network for Reconfigurable High Performance Computing,” in *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, 2018, pp. 192–201.

[86] N. Fujita, R. Kobayashi, Y. Yamaguchi, and T. Boku, “Parallel Processing on FPGA Combining Computation and Communication in OpenCL Programming,” in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2019, pp. 479–488.

[87] H. M. Waidyasooriya and M. Hariyama, “Multi-FPGA Accelerator Architecture for Stencil Computation Exploiting Spacial and Temporal Scalability,” *IEEE Access*, vol. 7, pp. 53 188–53 201, 2019.

[88] A. George, H. Lam, and G. Stitt, “Novo-G: At the Forefront of Scalable Reconfigurable Supercomputing,” *Computing in Science & Engineering*, vol. 13, no. 1, pp. 82–86, 2010.

[89] “IBM SuperVessel, OpenPower Cloud,” https://www.research.ibm.com/labs/china/supervessel.html [Last accessed: April 29, 2021].

[90] FaBRIC, “FPGA Accelerator Research Infrastructure Cloud (FAbRIC),” https://wikis.utexas.edu/display/fabric/Home [Last accessed: April 29, 2021].

[91] J. Weerasinghe, R. Polig, F. Abel, and C. Hagleitner, “Network-Attached FPGAs for Data Center Applications,” in *2016 International Conference on Field-Programmable Technology (FPT)*, 2016, pp. 36–43.

[92] J. Weerasinghe, F. Abel, C. Hagleitner, and A. Herkersdorf, “Disaggregated FPGAs: Network Performance Comparison against Bare-Metal Servers, Virtual Machines and Linux Containers,” in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 9–17.

[93] F. Abel, J. Weerasinghe, C. Hagleitner, B. Weiss, and S. Paredes, “An FPGA Platform for Hyperscalers,” in *2017 IEEE 25th Annual Symposium on High-Performance Interconnects (HOTI)*. IEEE, 2017, pp. 29–32.

[94] “Open Cloud Testbed-Exploring Next-Generation Cloud Platforms,” https://massopen.cloud/connected-initiatives/open-cloud-testbed/ [Last accessed: August 12, 2021].

[95] S. Handagal, M. Herbordt, and M. Leeser, “OCT: The Open Cloud FPGA Testbed,” in *31st International Conference on Field Programmable Logic and Applications (FPL)*, 2021.

[96] N. Tarafdar, T. Lin, E. Fukuda, H. Bannazadeh, A. Leon-Garcia, and P. Chow, “Enabling flexible network fpga clusters in a heterogeneous cloud data center,” in *FPGA*, 2017, pp. 237–246.

[97] N. Tarafdar, N. Eskandari, T. Lin, and P. Chow, “Designing for FPGAs in the Cloud,” *IEEE Design and Test*, vol. 35, no. 1, pp. 23–29, 2017.

[98] “Enzian,” http://enzian.systems/ [Last accessed: April 29, 2021].

[99] G. Alonso, T. Roscoe, D. Cock, M. Owaida, K. Kara, D. Korolija, Z. Wang *et al.*, “Tackling hardware/software co-design from a database perspective,” in *Proceedings of the 6th biennial Conference on Innovative Data Systems Research (CIDR), Amsterdam, Netherlands, January 2020.*, 2020.

[100] C. Conger, I. Troxel, D. Espinoza, V. Aggarwal, and A. George, “NARC: Network Attached Reconfigurable Computing for High Performance, Network Based Applications,” in *Proceedings of the Eighth Annual International Conference on Military and Aerospace Programmable Logic Devices (MAPLD’05)*, 2005.

[101] Cisco, “About the Nexus 3550-T Triton,” https://exablaze.com/docs/3550t/about/ [Last accessed: April 29, 2021].

[102] TheNextPlatform, “A Deep Dive into Cisco’s Use of Merchant Switch Chips,” https://www.nextplatform.com/ 2018/06/20/a-deep-dive-into-ciscos-use-of-merchant-switch-chips/ [Last accessed: April 29, 2021].

978-1-6654-2369-4/21/$31.00 ©2021 IEEE