Serverless FPGA
Work-In-Progress

Author(s):
Maschi, Fabio; Korolija, Dario; Alonso, Gustavo

Publication date:
2023-05-08

Permanent link:
https://doi.org/10.3929/ethz-b-000610035

Originally published in:
https://doi.org/10.1145/3592533.3592804

Serverless FPGA

Work-In-Progress

Fabio Maschi
Systems Group, Department of Computer Science, ETH Zurich
Zürich, Switzerland

Dario Korolija
Systems Group, Department of Computer Science, ETH Zurich
Zürich, Switzerland

Gustavo Alonso
Systems Group, Department of Computer Science, ETH Zurich
Zürich, Switzerland

ABSTRACT

In this short paper we investigate the combination of two emerging technologies: the tight provisioning requirements of Serverless computing and the acceleration potential of FPGAs. Serverless platforms suffer from container overheads, notably cold start latency, while having to adapt to Function-as-a-Service (FaaS) workloads. By exploring re-configurability of FPGAs and their acceleration power, we propose an innovative light-weight Serverless platform for FPGA-based FaaS applications which aims to reduce these overheads. In this study, we explore the feasibility of the idea by implementing key elements of such platform onto the FPGA. Our initial results show potential for acceleration in all aspects of function invocation.

ACM Reference Format:
Fabio Maschi, Dario Korolija, and Gustavo Alonso. 2023. Serverless FPGA: Work-In-Progress. In *The 1st Workshop on SErverless Systems, Applications and MEthodologies (SESAME '23), May 8, 2023, Rome, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3592533.3592804

1 INTRODUCTION

Serverless has become a popular and promising cloud computing paradigm. On the one hand, users benefit from on-demand costs and freedom from deployment concerns. On the other, cloud service providers enjoy fine-grained workloads that maximise computing hardware utilisation. Yet, achieving the perception of infinite resources and *serverless* comes at a cost. Serverless system infrastructure typically relies on existing platforms for containerised application, e.g., Kubernetes, Apache OpenWhisk and μVMs [1, 12, 16, 24]. However, such platforms were not originally designed to operate with the same set of requirements as the ones of serverless. For instance, containerised applications are more likely to be executed in a time window of hours or even days [20], compared to FaaS workloads that tipically have much shorter execution times on the order of 1 second or even just a few hundred milliseconds [24]. The provisioning of hundreds of thousands of short-lived serverless functions creates an unprecedented context switching overhead to the underlying system architecture. Moreover, the nature of applications leveraging serverless functions makes the workload inevitably latency-sensitive. In traditional containerised applications, the latency caused by cold-start delays, which can be tens of seconds or even minutes, is too significant for serverless functions, which should ideally be started in the microsecond range at most [24, 28].

At the same time, heterogeneous hardware architectures are becoming more available, with Field Programmable Gate Array (FPGAs) as a prominent option [3, 22]. FPGAs provide a spatial hardware architecture that operates under a much lower clock frequency (in MHz) compared to CPUs and GPUs (in GHz), but are designed to efficiently perform data processing in a multi-instruction-multi-data fashion. Besides providing great acceleration potential [2, 4, 5, 9, 17, 19, 25], one key advantage of FPGAs in comparison to more traditional accelerators is the dynamic ability to reconfigure portions of their computing fabric during runtime. More than that, FPGAs can be designed to partially re-configure only certain portions of their fabric without affecting the operation of the rest of the system [13, 21]. This provides the basis for multi-tenancy, necessary for the efficient utilisation of large computing resources in modern FPGAs, particularly in the context of cloud computing. In turn, this enables a completely new model of FPGA deployment, fit for serverless applications, characterised by dynamic accelerator libraries which could be *provisioned* on-demand, where and when needed.

In this work-in-progress paper, we present an initial exploration of such a serverless deployment platform integrated directly on an FPGA board. The FPGA runs an HTTP stack, which exposes the devices and its kernels over the network. Specific hardware-accelerated kernels can be made available as a FaaS registry, and be invoked over a RESTful interface. Preliminary results show the potential of such a design: moving the network (TCP/IP and HTTP stacks) to hardware provides orders of magnitude lower latency, and the platform can operate at much higher throughput compared to conventional, commercial HTTP servers running on CPUs. On top of that, partial reconfiguration, which directly correlates to function cold-start delays, is more deterministic, does not suffer from scalability problems, and is orders of magnitude faster on the FPGA than on a containerised stack.

2 BACKGROUND

Field-Programmable Gate Array (FPGA) is a matrix of logic elements that can be reconfigured to create various circuitry [27]. The spatial architecture of an FPGA allows custom designs to exploit parallelism through deep pipelining and concurrent instances of processing elements, which enables them to easily process the data at modern network line rates with minimum added latency overheads. Recent work have demonstrated advantages of exposing FPGA devices directly over the network [7, 8, 10, 26]. Combined

SESAME '23, May 8, 2023, Rome, Italy

ACM ISBN 979-8-4007-0185-6/23/05…$15.00
https://doi.org/10.1145/3592533.3592804

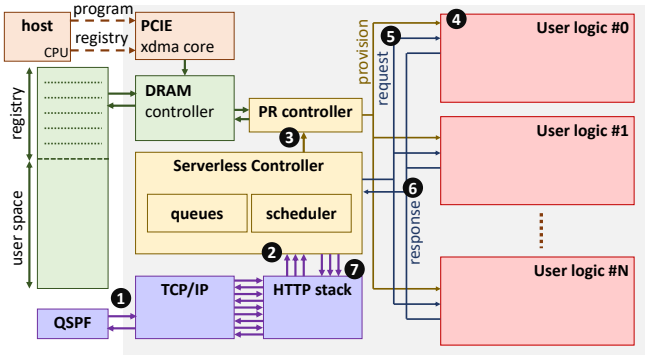

Figure 1: Block diagram of the architecture in the FPGA.

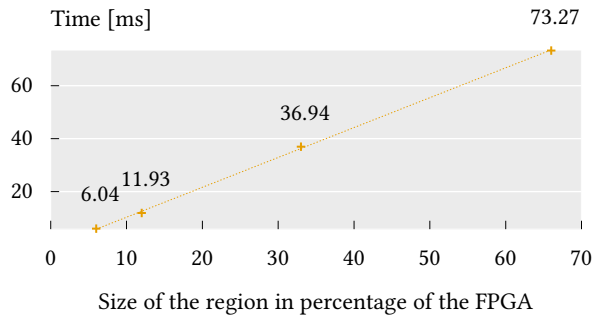

Figure 2: FPGA re-configuration latency in Coyote [15].

with Coyote [15], an FPGA shell that provides OS-like abstractions — notably multi-tenancy and context switching — FPGAs can become powerful and flexible processing nodes that could be exploited to leverage serverless workload requirements.

Du et al. [6] propose a complete framework for serverless computing to support accelerators (GPUs and FPGAs). They extend their platform running on the CPU with the capability of provisioning kernels on-demand. In this work, we make a serverless prototype module running standalone on the FPGA by exposing the system as a RESTful interface over HTTP, in such a manner that the accelerator can dynamically evolve and *provision* itself at runtime on-demand on incoming serverless requests. We combine partial re-configuration for multi-tenancy with function provisioning, an operation handled directly in the execution node (i.e., the FPGA), in contrast to the centralised Auto-scaling and Master Nodes used in traditional serverless platforms.

3 DESIGN

Figure 1 depicts the system architecture of our serverless platform running on the FPGA. It is important to highlight that the host CPU, connected through the PCIe bus to the FPGA, is only involved during provisioning of the platform, when it programs the FPGA with the shell and loads the registry of functions in its local memory. This operation is equivalent to provisioning the Kubernetes platform itself (i.e., Controller Manager, Master Node, Scheduler, etc) in a traditional CPU deployment, and does not interfere with the life cycle of a serverless function. The registry holds all available functions that can be stored in the FPGA. As a first prototype, we store it in the local FPGA memory. Real-world deployments would use a distributed storage system, that can be either fetched via RDMA from another node of the system, or from the network [26]. HTTP requests carry key information to determine the function to be executed, data payload and meta values for the platform.

The execution of a serverless function is enumerated in Figure 1: ❶ the request comes from the network, received by the TCP/IP module and parsed by the HTTP stack; ❷ the Serverless Controller processes the function, allocating memory in the local DRAM and buffering the request in internal queues; ❸ the scheduler calls the Partial Re-configurable Controller to load the function to be executed from the registry; ❹ the PR Controller provisions a sub-region according to the scheduler; ❺ the HTTP request is forwarded to the sub-region and executed; ❻ the user logic returns the result of the function execution; ❼ the Serverless Controller sets the sub-region as idle, and issues the HTTP response back to the client.

The combination of several FPGA nodes would compose a serverless FPGA cluster. We envisage that, as the entry point of such a system, a basic router node would be responsible of load balancing requests to the set of execution nodes. Given that each execution node is fully autonomous and has its own Controller, this load balancing can initially be as simple as using a round-robin distribution. By not requiring complex HTTP parsing or expensive scheduling algorithms, the latency introduced by this module is likely to be limited to the microsecond range, primarily cause by network round-trip latency [8].

4 EVALUATION

In this section we show two micro-benchmarks, the first demonstrating the negligible overheads of partial re-configuration and the second the performance advantages of a fully-offloaded HTTP stack to the FPGA.

Figure 2 shows the time taken to partially re-configure the sub-regions of the FPGA with user logic. The operation presents a linear response as a function to the size of the sub-region. This can be mapped to VM provisioning in the context of FPGA kernels. Differently from cold starts and congestion caused by propagation scaling decisions from the Master node in container-based platforms, the FPGA implementation is able to sustain invariable response time for such operation, as control logic is handled internally in hardware, and memory bandwidth is big enough for such utilisation. The magnitude of waiting and cold start delays in current Serverless computing platforms [23] is an order of magnitude higher than the one observed in FPGA context switching. This implies that the overheads of such context switching would be negligible in comparison to standard serverless cold start times while at the same time enabling the full acceleration potential of modern FPGAs.

We performed an evaluation of our HTTP server stack running on the FPGA and compared it to an open source commercial engine nginx [18] running on the CPU. The purpose of this is primarily to evaluate the FPGA's ability to handle HTTP request-responses in comparison to a CPU-based approach, with the aim of assessing whether hosting the HTTP server on the host CPU and forwarding requests to the FPGA over PCIe is a suitable option or not. In Figure 3 the end-to-end latency distribution of 5000 sequentially

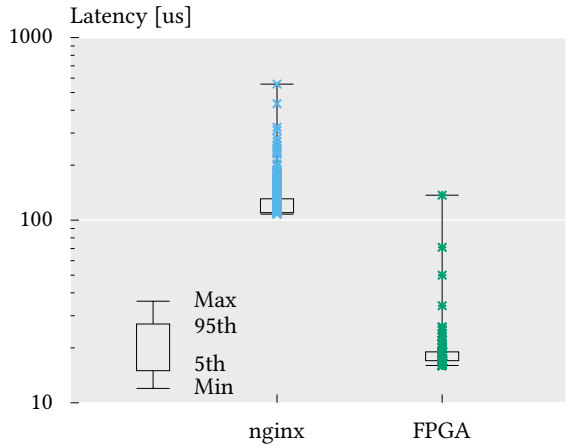

Figure 3: Latency distribution of 5000 sequential HTTP requests as seen from a CPU client.

issued request-responses by a single CPU HTTP client is presented, along with the maximum, minimum, and 95th and 5th percentiles. The FPGA exhibits significantly lower latency than nginx, with execution times up to an order of magnitude faster, as low as 16 µs in contrast to 108 µs on the CPU. Notably, the FPGA displays very low sample variance, with only 4 out of 5000 measurements having a latency greater than 30 µs, and the gap between the minimum and 95th percentile being only 3 µs, compared to 23 µs for nginx. The slowest points for both sets is caused by the first request in each experiment needing to establish a TCP/IP session.

5 CONCLUSION

In this paper, we discuss the idea of FPGA-based applications exposed via a RESTful interface as functions in a serverless setting. Preliminary results show significant performance advantages in the architecture due to the much lower latency in reacting to requests, as well as the ability to process the request directly on the FPGA, without the intervention of a host CPU.

Next steps consist of finishing the hardware implementation of the scheduler, notably exploring different scheduling algorithms for multi-tenancy, and optimising the hardware queues so that they can cope with different workloads. In addition, we are also integrating the FPGA-based serverless platform with a set of heterogeneous applications running on a cluster of FPGAs in order to exploit dynamic scalability to tens of FPGA nodes [11, 14]. Finally, the current Registry is planned to be moved from the FPGA internal memory to a remote memory device, allowing the system to store and consume a large set of deployed functions, required to achieve the scale of serverless deployments.

REFERENCES

[1] Alexandru Agache, Marc Brooker, Alexandra Iordache, Anthony Liguori, Rolf Neugebauer, Phil Piwonka, and Diana-Maria Popa. 2020. Firecracker: Lightweight Virtualization for Serverless Applications. In *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, Ranjita Bhagwan and George Porter (Eds.). USENIX Association, 419–434. https://www.usenix.org/conference/nsdi20/presentation/agache

[2] Ahmed Ghazi Blaiech, Khaled Ben Khalifa, Carlos Valderrama, Marcelo A. C. Fernandes, and Mohamed Hedi Bedoui. 2019. A Survey and Taxonomy of FPGA-based Deep Learning Accelerators. *J. Syst. Archit.* 98 (2019), 331–345. https://doi.org/10.1016/j.sysarc.2019.01.007

[3] Christophe Bobda, Joel Mandebi Mbongue, Paul Chow, Mohammad Ewais, Naif Tarafdar, Juan Camilo Vega, Ken Eguro, Dirk Koch, Suranga Handagala, Miriam Leeser, Martin C. Herbordt, Hafsah Shahzad, Peter Hofste, Burkhard Ringlein, Jakub Szefer, Ahmed Sanaullah, and Russell Tessier. 2022. The Future of FPGA Acceleration in Datacenters and the Cloud. *ACM Trans. Reconfigurable Technol. Syst.* 15, 3 (2022), 34:1–34:42. https://doi.org/10.1145/3506713

[4] Monica Chiosa, Fabio Maschi, Ingo Müller, Gustavo Alonso, and Norman May. 2022. Hardware Acceleration of Compression and Encryption in SAP HANA. *Proc. VLDB Endow.* 15, 12 (2022), 3277–3291. https://www.vldb.org/pvldb/vol15/p3277-chiosa.pdf

[5] Eric S. Chung, Jeremy Fowers, Kalin Ovtcharov, Michael Papamichael, Adrian M. Caulfield, Todd Massengill, Ming Liu, Daniel Lo, Shlomi Alkalay, Michael Haselman, Maleen Abeydeera, Logan Adams, Hari Angepat, Christian Boehn, Derek Chiou, Oren Firestein, Alessandro Forin, Kang Su Gatlin, Mahdi Ghandi, Stephen Heil, Kyle Holohan, Ahmad El Husseini, Tamás Juhász, Kara Kagi, Ratna Kovvuri, Sitaram Lanka, Friedel van Megen, Dima Mukhortov, Prerak Patel, Brandon Perez, Amanda Rapsang, Steven K. Reinhardt, Bita Rouhani, Adam Sapek, Raja Seera, Sangeetha Shekar, Balaji Sridharan, Gabriel Weisz, Lisa Woods, Phillip Yi Xiao, Dan Zhang, Ritchie Zhao, and Doug Burger. 2018. Serving DNNs in Real Time at Datacenter Scale with Project Brainwave. *IEEE Micro* 38, 2 (2018), 8–20. https://doi.org/10.1109/MM.2018.022071131

[6] Dong Du, Qingyuan Liu, Xueqiang Jiang, Yubin Xia, Binyu Zang, and Haibo Chen. 2022. Serverless computing on heterogeneous computers. In *ASPLOS '22: 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 28 February 2022 - 4 March 2022*, Babak Falsafi, Michael Ferdman, Shan Lu, and Thomas F. Wenisch (Eds.). ACM, 797–813. https://doi.org/10.1145/3503222.3507732

[7] Daniel Firestone, Andrew Putnam, Sambrama Mundkur, Derek Chiou, Alireza Dabagh, Mike Andrewartha, Hari Angepat, Vivek Bhanu, Adrian M. Caulfield, Eric S. Chung, Harish Kumar Chandrappa, Somesh Chaturmohta, Matt Humphrey, Jack Lavier, Norman Lam, Fengfen Liu, Kalin Ovtcharov, Jitu Padhye, Gautham Popuri, Shachar Raindel, Tejas Sapre, Mark Shaw, Gabriel Silva, Madhan Sivakumar, Nisheeth Srivastava, Anshuman Verma, Qasim Zuhair, Deepak Bansal, Doug Burger, Kushagra Vaid, David A. Maltz, and Albert G. Greenberg. 2018. Azure Accelerated Networking: SmartNICs in the Public Cloud. In *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, Sujata Banerjee and Srinivasan Seshan (Eds.). USENIX Association, 51–66. https://www.usenix.org/conference/nsdi18/presentation/firestone

[8] Zhenhao He, Dario Korolija, and Gustavo Alonso. 2021. EasyNet: 100 Gbps Network for HLS. In *31st International Conference on Field-Programmable Logic and Applications, FPL 2021, Dresden, Germany, August 30 - Sept. 3, 2021*. IEEE, 197–203. https://doi.org/10.1109/FPL53798.2021.00040

[9] Gui Huang, Xuntao Cheng, Jianying Wang, Yujie Wang, Dengcheng He, Tieying Zhang, Feifei Li, Sheng Wang, Wei Cao, and Qiang Li. 2019. X-Engine: An Optimized Storage Engine for Large-scale E-commerce Transaction Processing. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 651–665. https://doi.org/10.1145/3299869.3314041

[10] Zsolt István, David Sidler, and Gustavo Alonso. 2017. Caribou: Intelligent Distributed Storage. *Proc. VLDB Endow.* 10, 11 (2017), 1202–1213. https://doi.org/10.14778/3137628.3137632

[11] Wenqi Jiang, Zhenhao He, Shuai Zhang, Kai Zeng, Liang Feng, Jiansong Zhang, Tongxuan Liu, Yong Li, Jingren Zhou, Ce Zhang, and Gustavo Alonso. 2021. FleetRec: Large-Scale Recommendation Inference on Hybrid GPU-FPGA Clusters. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 3097–3105. https://doi.org/10.1145/3447548.3467139

[12] Knative. 2023. *Knative.* https://knative.dev

[13] Oliver Knodel, Paul R Genssler, Fredo Erxleben, and Rainer G Spallek. 2018. FPGAs and the cloud–An endless tale of virtualization, elasticity and efficiency. *International Journal on Advances in Systems and Measurements* 11, 3-4 (2018), 230–249.

[14] Dario Korolija, Dimitrios Koutsoukos, Kimberly Keeton, Konstantin Taranov, Dejan S. Milojicic, and Gustavo Alonso. 2022. Farview: Disaggregated Memory with Operator Off-loading for Database Engines. In *12th Conference on Innovative Data Systems Research, CIDR 2022, Chaminade, CA, USA, January 9-12, 2022*. www.cidrdb.org. https://www.cidrdb.org/cidr2022/papers/p11-korolija.pdf

[15] Dario Korolija, Timothy Roscoe, and Gustavo Alonso. 2020. Do OS abstractions make sense on FPGAs?. In *14th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2020, Virtual Event, November 4-6, 2020*. USENIX Association, 991–1010. https://www.usenix.org/conference/osdi20/presentation/roscoe

[16] Kubernetes. 2023. *Kubernetes.* https://kubernetes.io

[17] Fabio Maschi, Muhsen Owaida, Gustavo Alonso, Matteo Casalino, and Anthony Hock-Koon. 2020. Making Search Engines Faster by Lowering the Cost of Querying Business Rules Through FPGAs. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 2255–2270. https://doi.org/10.1145/3318464.3386133

[18] nginx. 2023. *nginx open source.* https://nginx.org

[19] Muhsen Owaida, Gustavo Alonso, Laura Fogliarini, Anthony Hock-Koon, and Pierre-Etienne Melet. 2019. Lowering the Latency of Data Processing Pipelines Through FPGA based Hardware Acceleration. *Proc. VLDB Endow.* 13, 1 (2019), 71–85. https://doi.org/10.14778/3357377.3357383

[20] Chunyi Peng, Minkyong Kim, Zhe Zhang, and Hui Lei. 2012. VDN: Virtual machine image distribution network for cloud data centers. In *Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25-30, 2012*, Albert G. Greenberg and Kazem Sohraby (Eds.). IEEE, 181–189. https://doi.org/10.1109/INFCOM.2012.6195556

[21] Burkhard Ringlein, François Abel, Alexander Ditter, Beat Weiss, Christoph Hagleitner, and Dietmar Fey. 2019. System Architecture for Network-Attached FPGAs in the Cloud using Partial Reconfiguration. In *29th International Conference on Field Programmable Logic and Applications, FPL 2019, Barcelona, Spain, September 8-12, 2019*, Ioannis Sourdis, Christos-Savvas Bouganis, Carlos Álvarez, Leonel Antonio Toledo Díaz, Pedro Valero-Lara, and Xavier Martorell (Eds.). IEEE, 293–300. https://doi.org/10.1109/FPL.2019.00054

[22] Robert Schmid, Max Plauth, Lukas Wenzel, Felix Eberhardt, and Andreas Polze. 2020. Accessible near-storage computing with FPGAs. In *EuroSys '20: Fifteenth EuroSys Conference 2020, Heraklion, Greece, April 27-30, 2020*, Angelos Bilas, Kostas Magoutis, Evangelos P. Markatos, Dejan Kostic, and Margo I. Seltzer (Eds.). ACM, 28:1–28:12. https://doi.org/10.1145/3342195.3387557

[23] Mohammad Shahrad, Jonathan Balkind, and David Wentzlaff. 2019. Architectural Implications of Function-as-a-Service Computing. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2019, Columbus, OH, USA, October 12-16, 2019.* ACM, 1063–1075. https://doi.org/10.1145/3352460.3358296

[24] Mohammad Shahrad, Rodrigo Fonseca, Iñigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, and Ricardo Bianchini. 2020. Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider. In *2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, Ada Gavrilovska and Erez Zadok (Eds.). USENIX Association, 205–218. https://www.usenix.org/conference/atc20/presentation/shahrad

[25] David Sidler, Zsolt István, Muhsen Owaida, and Gustavo Alonso. 2017. Accelerating Pattern Matching Queries in Hybrid CPU-FPGA Architectures. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu (Eds.). ACM, 403–415. https://doi.org/10.1145/3035918.3035954

[26] David Sidler, Zeke Wang, Monica Chiosa, Amit Kulkarni, and Gustavo Alonso. 2020. StRoM: smart remote memory. In *EuroSys '20: Fifteenth EuroSys Conference 2020, Heraklion, Greece, April 27-30, 2020*, Angelos Bilas, Kostas Magoutis, Evangelos P. Markatos, Dejan Kostic, and Margo I. Seltzer (Eds.). ACM, 29:1–29:16. https://doi.org/10.1145/3342195.3387519

[27] Jens Teubner and Louis Woods. 2013. *Data Processing on FPGAs.* Morgan & Claypool Publishers. https://doi.org/10.2200/S00514ED1V01Y201306DTM035

[28] Ao Wang, Shuai Chang, Huangshi Tian, Hongqi Wang, Haoran Yang, Huiba Li, Rui Du, and Yue Cheng. 2021. FaaSNet: Scalable and Fast Provisioning of Custom Serverless Container Runtimes at Alibaba Cloud Function Compute. In *2021 USENIX Annual Technical Conference, USENIX ATC 2021, July 14-16, 2021*, Irina Calciu and Geoff Kuenning (Eds.). USENIX Association, 443–457. https://www.usenix.org/conference/atc21/presentation/wang-ao