System Software Support for Fast and Flexible Task Management on a Large-scale FPGA cluster

Atsushi Koshiba, Kentaro Sano
RIKEN Center for Computational Science
Kobe, Hyogo, Japan
{atsushi.koshiba,kentaro.sano}@riken.jp

EXTENDED ABSTRACT

Hardware acceleration using Field Programmable Gate Arrays (FPGAs) has been receiving much attention in the field of high performance computing (e.g., data centers and supercomputers), where not only computational performance but also power efficiency is highly required due to their energy constraints. FPGA provides a reconfigurable region where users implement custom hardware specialized to accelerate a specific task. The custom hardware on FPGA has the potential to achieve several tens of times performance improvement compared to CPUs/GPUs. FPGA-based large-scale computing is getting popular (e.g., Bing search engine [2]).

Although FPGA is a promising technology to outperform commodity computer systems, system software support for a large-scale FPGA cluster is still early-stage. One of the challenges is scalability. FPGA applications are commonly written in OpenCL programming model with vendor-specific development tools (e.g., Intel OpenCL SDK and SDAccel). The OpenCL programming model and development tools guarantee high programmability, while they only support single-node execution and the applications do not scale up to a multi-node FPGA cluster. In addition, existing systems are lacking in supporting FPGA resource sharing. Large-scale computer systems such as supercomputers and cloud servers are shared among many users and various types of applications. To efficiently execute numerous applications of different users with a limited amount of FPGA resource, resource isolation and load balancing systems are essential.

To achieve high scalability, programmability and efficient FPGA sharing among users, we propose a hypervisor-based FPGA virtualization system that enables fast and flexible task management on a large-scale FPGA cluster. Figure 1 shows an overview of the proposed system. In the proposed system, OpenCL host applications are executed on a *unikernel*, a lightweight library OS that provides a minimal set of OS functions as a user library. The unikernel provides OpenCL-compatible APIs and OS device driver functions to manage FPGA resources. The hypervisor performs FPGA resource allocation and task scheduling based on application demands and run-time FPGA resource usage. The hypervisor can allocate FPGA resources on different nodes to unikernel applications according to their requests. Resource isolation and memory protection are realized by the hypervisor and hardware components (FPGA shell) so that the system can support FPGA sharing among applications. In addition, a load balancer in the hypervisor performs a fast live migration of unikernel applications among physical server machines/FPGAs. These functions achieve high scalability, high programmability, resource isolation, and high resource utilization with lower virtualization overhead.

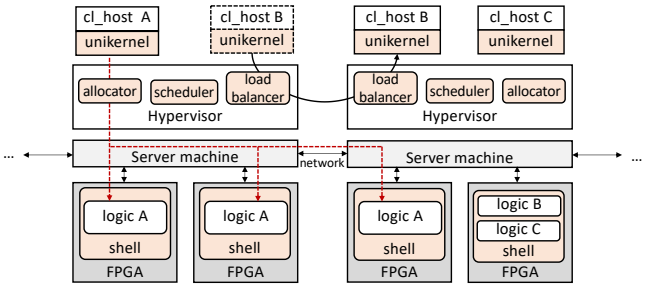

Figure 1: An overview of the proposed system.

System software support for large-scale computing with FPGAs has been studied in recent years. Blaze is a framework that provides programming and run-time support to deploy big data processing applications such as Hadoop in an FPGA-attached server cluster [3]. Unlike Blaze, our research focuses on supporting OpenCL applications implemented with commodity FPGA development tools. Another research proposes a cloud FPGA virtualization mechanism that allows virtual machines (VMs) to share reconfigurable regions of multiple FPGA boards through OpenStack [1]. In contrast to this VM-level approach, our system allows FPGAs to be shared by applications, which achieves finer-grained task allocation and flexible load balancing.

We are currently implementing a prototype of our system on a Xilinx Zynq SoC board. The base software of the proposed system such as a hypervisor (KVM) and a unikernel (Solo5) is already running on the Zynq board. We are also planning to implement our system in a realistic cloud server system (e.g., x86 server cluster with Alveo U250, a high-performance FPGA board).

ACKNOWLEDGMENTS

This research was supported by JSPS KAKENHI Grant Number JP19K24360.

REFERENCES

[1] S. Byma, J. G. Steffan, H. Bannazadeh, A. L. Garcia, and P. Chow. 2014. FPGAs in the Cloud: Booting Virtualized Hardware Accelerators with OpenStack. In *2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines*. 109–116. https://doi.org/10.1109/FCCM.2014.42

[2] Adrian M. Caulfield, Eric S. Chung, Andrew Putnam, Hari Angepat, Jeremy Fowers, Michael Haselman, Stephen Heil, Matt Humphrey, Puneet Kaur, Joo-Young Kim, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Papamichael, Lisa Woods, Sitaram Lanka, Derek Chiou, and Doug Burger. 2016. A Cloud-scale Acceleration Architecture. In *The 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-49)*. IEEE Press, Piscataway, NJ, USA, Article 7, 13 pages. http://dl.acm.org/citation.cfm?id=3195638.3195647

[3] Muhuan Huang, Di Wu, Cody Hao Yu, Zhenman Fang, Matteo Interlandi, Tyson Condie, and Jason Cong. 2016. Programming and Runtime Support to Blaze FPGA Accelerator Deployment at Datacenter Scale. In *Proceedings of the Seventh ACM Symposium on Cloud Computing (SoCC '16)*. ACM, New York, NY, USA, 456–469. https://doi.org/10.1145/2987550.2987569