Chair of Decentralized Systems Engineering
Department of Informatics

Hardware-assisted I/O Management for Cloud FPGAs

General information	**Advisor**	Dr. Atsushi Koshiba
	Email	atsushi.koshiba@tum.de
	Date	31.1.2022
Type	Master / Bachelor / Guided Research	

Description

Hardware acceleration with Field Programmable Gate Arrays (FPGAs) is a key technology to satisfy the high-performance demands of cloud workloads under the severe energy constraint in data centers. FPGAs are well suited for data processing as they can be connected directly to storage or network devices so that custom logics on FPGA process the incoming data without going through CPU-side memory. Such peer-to-peer communications can significantly improve the performance and energy efficiency of the computation.

Although FPGA has the advantage of accelerating data-intensive workloads in such ways, they are lack programmability and portability. FPGA vendors offer programming interfaces enabling peer-to-peer communications between FPGA and other devices [1, 2]. However, they are realized by vendor-specific pragmas or extensions and are only available on specified FPGA boards. Custom logics written in this manner are not portable among different FPGA vendors/board types.

LLVM compilers for High-Level Synthesis (HLS) [3] and Domain-Specific Language (DSL) [4] have been well studied for improving FPGA programmability and portability. They interpret software code written in C/C++ or other high-level languages into register transfer level (RTL) code that is synthesizable on the target FPGA. The compiler-based approaches are still not sufficient for I/O management because available I/O interfaces depend on the target FPGA platform; I/O interfaces such as PCIe bridges or network ports are statically integrated on FPGAs and not reconfigurable from the user side.

This project aims to build hardware-assisted I/O management for peer-to-peer communication on FPGAs. We propose an I/O abstraction layer called *Shell* bridging custom logics on the reconfigurable region and I/O ports on the static region. The Shell is responsible for abstracting I/O interfaces of user logics and multiplexing data transfer requests between the logics and the outside devices. The proposed Shell will be implemented upon Coyote [5], an open-sourced FPGA Shell. We will also verify the portability and performance of the Shell on various FPGA cards we have.

Keywords

FPGA, LLVM, heterogeneous computing, portability, near-data processing, networking

Goals

Concrete outcomes

1. Design an FPGA Shell that abstracts the communication layer for FPGA custom logic.
2. Implement the compiler by extending the Coyote Shell [5].
3. Evaluate the portability and performance of the Shell on two different Xilinx FPGA cards: Alveo U50 and Nexys Video.

Bonus points

1. Publish the Shell as an open-sourced project.
2. Support an FPGA card from different vendors, i.e., Intel FPGA.

Prerequisites

Compulsory

- Experience in FPGA programming (HDL, HLS, etc.)

Preferred

- Experience in FPGA CAD tools (Quartus, Vivado, Vitis, etc.)
- Knowledge and experience of C/C++ coding

- Knowledge of heterogeneous computing (CUDA, OpenCL, etc.)

References

1. PCIe Peer-to-Peer, https://xilinx.github.io/XRT/master/html/p2p.html
2. Intel FPGA SDK for OpenCL Overview, https://www.intel.com/content/www/us/en/programmable/documentation/mwh1391807965224.html#ewa1411747396740
3. A Survey and Evaluation of FPGA High-Level Synthesis Tools, IEEE Trans on CADICS’16, https://ieeexplore.ieee.org/abstract/document/7368920
4. Generating FPGA-based image processing accelerators with Hipacc, ICCAD’17, https://ieeexplore.ieee.org/document/8203894
5. Do OS abstractions make sense on FPGAs?, OSDI’20, https://dl.acm.org/doi/abs/10.5555/3488766.3488822

Application process

Please send an email to the advisor including the following:

- Email subject: “Thesis application (DSE)”
- CV
- A copy of your transcript(s)
- A **motivation statement**, please include samples of your work that you are proud of (e.g., major projects, open-source contributions, Github page, etc.) and/or writing samples (e.g., your technical blog, project reports, etc.)