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

Atsushi Koshiba, Kentaro Sano

RIKEN Center for Computational Science (R-CCS)

Background

High-performance Computing with FPGAs

- ✓ FPGAs scale the performance for specific tasks more than CPUs
  - e.g., stencil computation, machine learning, FFT
- ✓ Large-scale FPGA clusters are getting more popular in cloud/HPC:
  - Data center : Amazon EC2 F1 (Amazon AWS) [1]
  - Supercomputers : Cygnus (Tsukuba Univ.) [2], FPGA cluster (RIKEN) [3]

FPGA sharing among users/apps are challenging

- ✓ Shared among different users and massive applications
  → data isolation, load balancing, performance scaling are essential

Instance types of Amazon EC2 F1 [1]

Name	FPGAs	vCPUs	memory	Price/hour
f1.2xlarge	1	8	122 GiB	$1.65
f1.16xlarge	8	64	976 GiB	$13.20

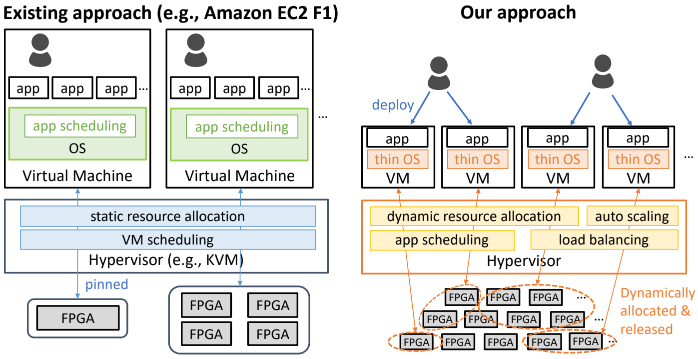
Cygnus [2]

Challenge & Solution

System software support for FPGA clusters is early-stage

- ✓ Existing work: FPGAs are **statically** allocated to each VM instance
  → large overhead, low resource usage, low scalability
- ✓ Our proposal: FPGAs are **dynamically** allocated to each application
  → low overhead, flexible load balancing & auto scaling

Existing approach (e.g., Amazon EC2 F1) **Our approach**



Types of Virtualized Environments

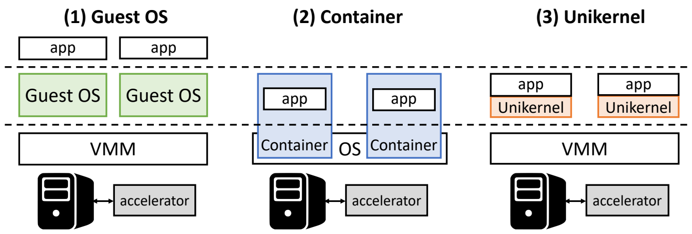
Requirements for Virtualized Execution Environments

- ✓ *Isolation* : each app is isolated and not interfered each other
- ✓ *Virt. overhead* : I/O control overhead should be avoided
- ✓ *Programmability* : existing app code (e.g., OpenCL) is runnable
- ✓ *Elasticity* : auto scaling and load balancing

Unikernels are suitable for FPGA virtualization

- ✓ Guest OS : Secure but high virtualization overhead
- ✓ Container : low overhead but less secure than Guest OS
- ✓ Unikernel : Secure (isolated) and low-overhead

→ Propose a unikernel-based FPGA virtualization system & a mechanism to support programmability and elasticity



[1] Amazon AWS, Amazon EC2 F1 Instances, https://aws.amazon.com/ec2/instance-types/f1/?nc1=h_ls
[2] Tsukuba University, Cygnus, https://www.ccs.tsukuba.ac.jp/eng/supercomputers/
[3] Miyajima+, "High-Performance Custom Computing with FPGA Cluster as an Off-loading Engine," SC'19, poster.

Proposed Mechanism

Fast and flexible FPGA virtualization using unikernels

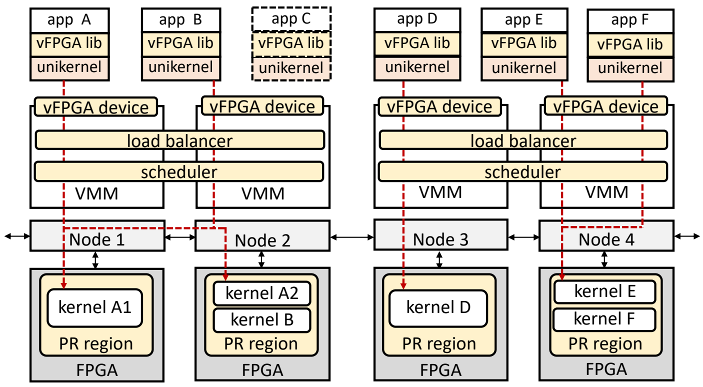
- ✓ Each app is running on its dedicated unikernel independently
- ✓ Hypervisor (VMM) mediates unikernel apps that require FPGAs

Para virtualization of FPGAs

- ✓ Unikernels provide heterogeneous programming libraries (e.g., OpenCL) that allows apps to manage virtualized FPGA devices
- ✓ Prevent unikernels from mapping the same FPGA address space
  →ensure isolation & programmability

Unikernel scheduler & Load balancer

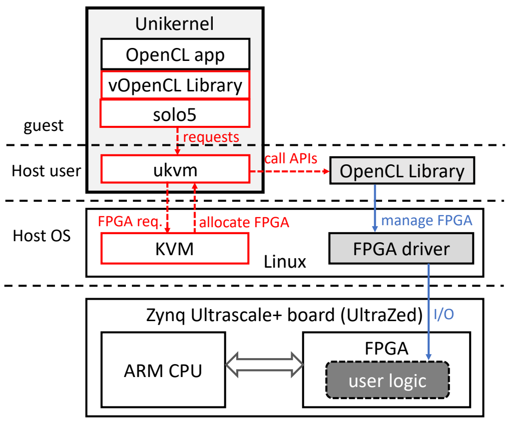
- ✓ VMM decides which nodes/FPGAs should be allocated to unikernels aps according to their demands and FPGA usage
- ✓ Partial Reconfiguration (PR) makes single FPGA sharable among apps
- ✓ Resource allocation is dynamically changed if a node or FPGA becomes more/less busy →ensure elasticity



Prototype

Implementing a prototype on Zynq Ultrascale+ SoC

- ✓ A hypervisor (KVM) and unikernel (solo5) are running on Zynq SoC
- ✓ Also planning to implement our system on a realistic cloud server system (x86 server with Alveo U250)



Future Work

Implementation and Evaluation of our system

- ✓ Evaluate baseline performance with Zynq SoC/Alveo card
- ✓ Consider applicability of our system to other accelerators (e.g., GPU)
- ✓ Implement realistic applications (e.g., FFT, fluid simulation)

Acknowledgments: This research was supported by JSPS KAKENHI Grant Number 19K24360.