Proteus

Heterogeneous FPGA Virtualization

Felix Gust, Shu Anzai, Charalampos Mainas, Atsushi Koshiba,
Pramod Bhatotia

Systems Research Group
https://dse.in.tum.de/

EuroSys 2026, Edinburgh, UK

FPGAs in the Cloud

- Increasing adoption of FPGAs in the cloud
- Advantage of FPGAs: Reconfigurability
  - Flexibility
  - Customizability
  - Performance
  - Energy efficiency

FPGA Heterogeneity

Cloud FPGAs are heterogeneous in logic, memory, runtime, performance, ...

MS Azure NP

AMD U250
3780k logic cells
64GB DDR4
Vitis/XRT

AWS EC2 F2

AMD VU47P
2852k logic cells
16GB HBM, 64GB DDR4
Vitis/XRT

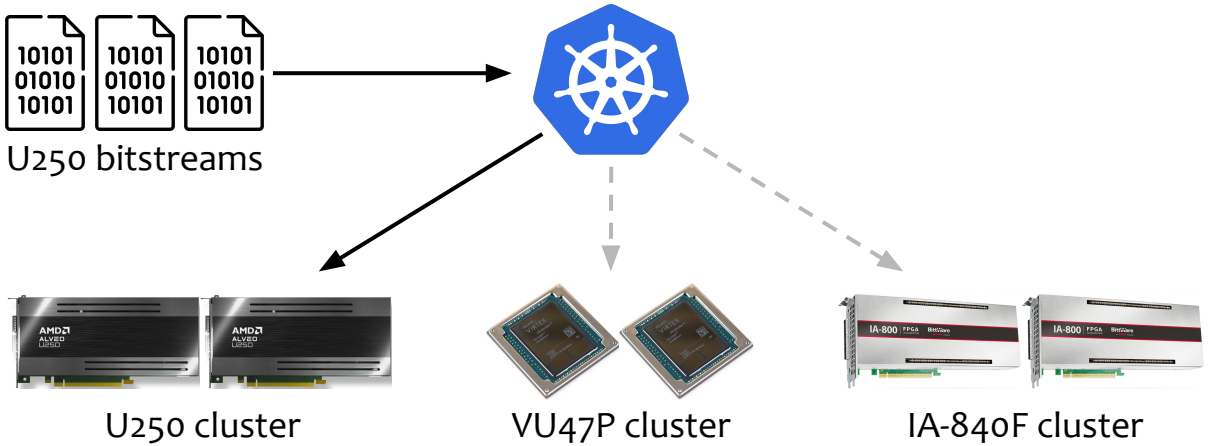
VMAccel

Intel IA-840F
2692k logic cells
32GB DDR4
oneAPI

The Problem

Heterogeneity forces users to choose between

- investing significant development effort to target multiple FPGA models
- target single FPGA model, leaving others unused



FPGA Heterogeneity

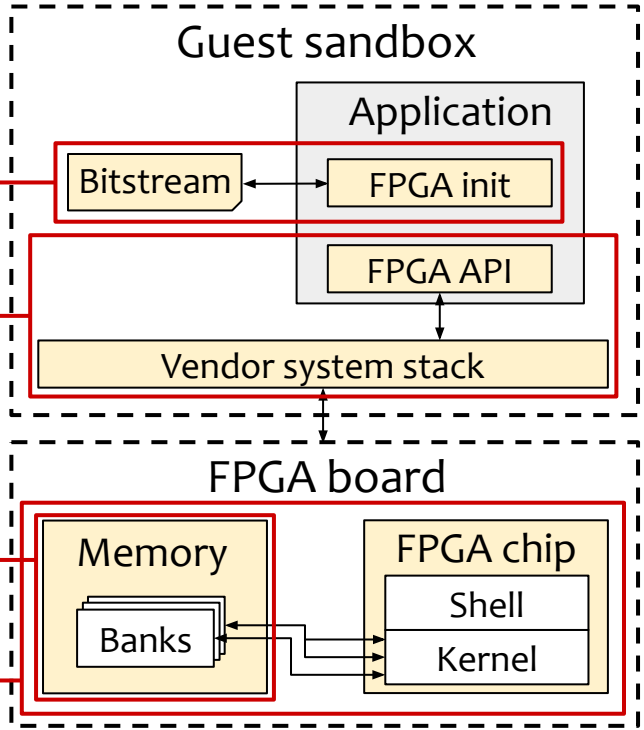
FPGA heterogeneity stems from four main factors:

#1 Bitstream incompatibility

#2 Vendor-specific APIs

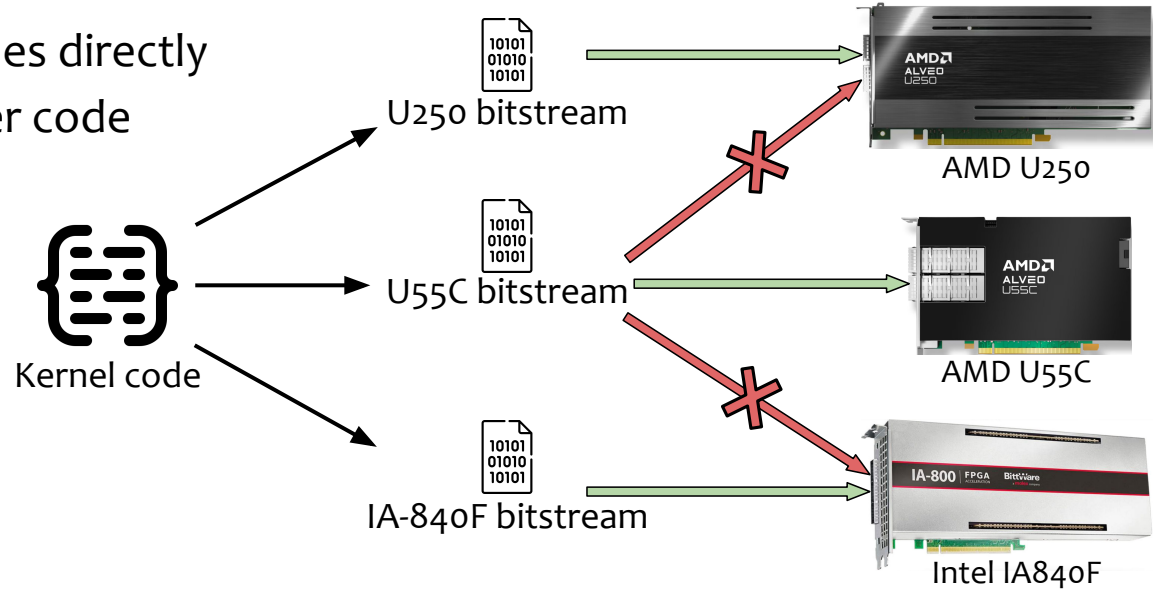
#3 Heterogeneous memory

#4 Performance variance



#1: Bitstream Incompatibility

FPGA APIs and runtimes directly expose devices to user code



Bitstream incompatibility and **lack of virtualization** limit application portability and compatibility

#2: Vendor-specific APIs

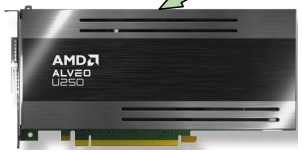
Existing APIs expose low-level device details

Host code

```
#include <cl_ext_xilinx.h>
...
cl_mem_ext_ptr_t ext = {0};
ext.flags = XCL_MEM_DDR_BANK0;
buf = cl::Buffer(CL_MEM_EXT_PTR_XILINX, size, &ext);
...
```

Vendor-specific code (AMD with DDR)

AMD XRT runtime



AMD U250

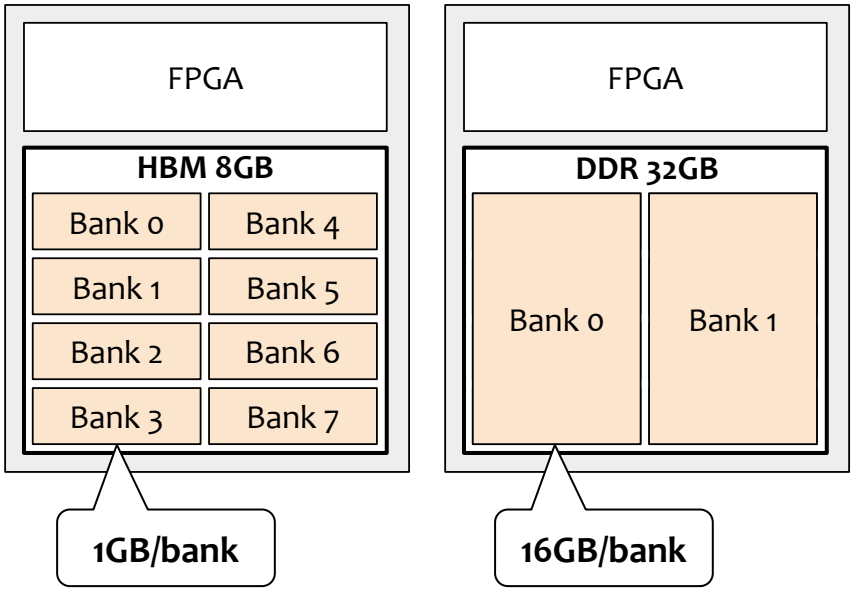
AMD U55C (no DDR)

Intel IA840F

API heterogeneity makes the host code incompatible across vendor-specific FPGA platforms

#3: Heterogeneous Memory

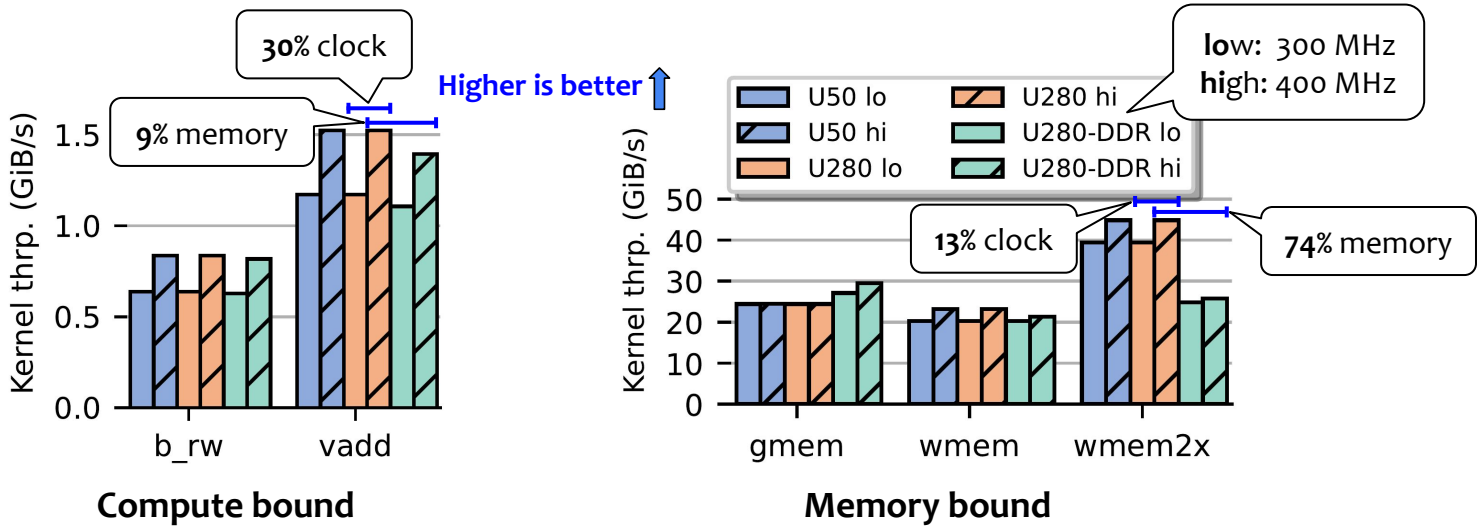
- Off-chip memory varying in type (DDR/HBM) and capacity
- Memory heterogeneity exposed to applications as static configuration or low-level APIs



Memory heterogeneity increases engineering effort and limits kernel compatibility

#4: Performance Variance

Main factors: kernel clock speed & off-chip memory



Unawareness of performance gaps between FPGAs leads to unexpected performance drops

Research Question

Can we design a new **cloud FPGA framework that virtualizes and orchestrates heterogeneous FPGAs** to improve utilization, flexibility, and performance?

Proteus

Proteus: Heterogeneous FPGA virtualization
A framework to virtualize and orchestrate heterogeneous FPGAs in the cloud

Contributions

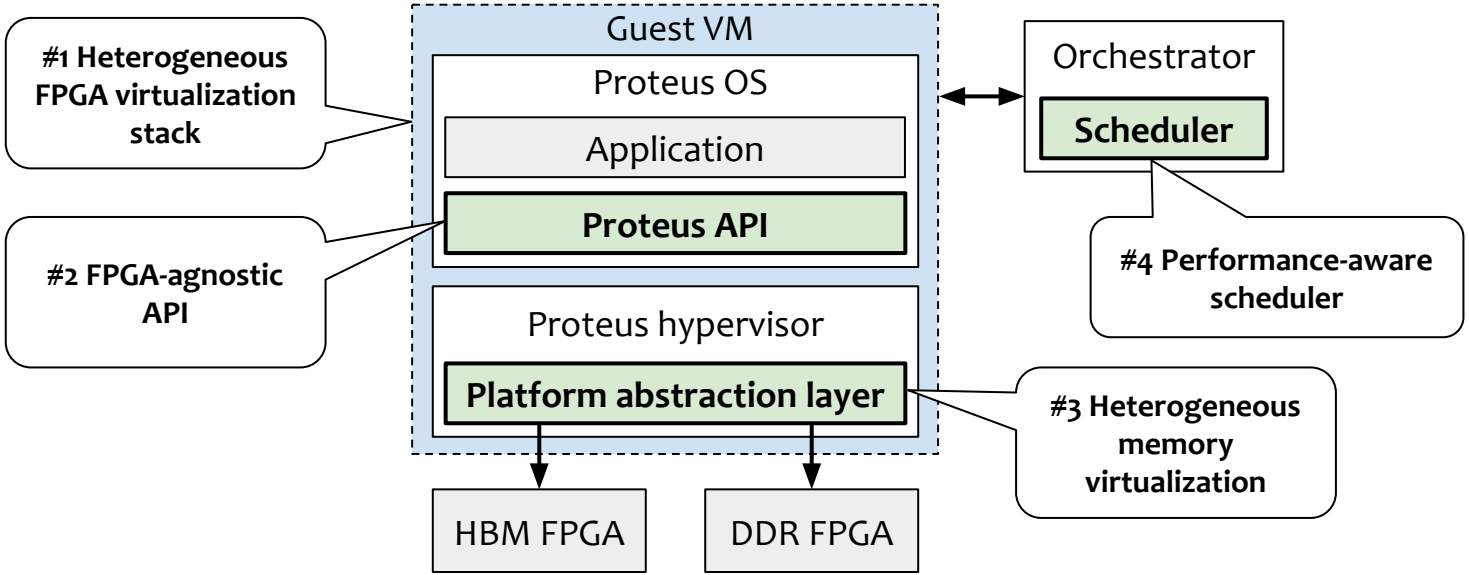
- Heterogeneous FPGA virtualization stack
- FPGA-agnostic API
- Heterogeneous memory virtualization
- Performance-aware scheduler

Outline

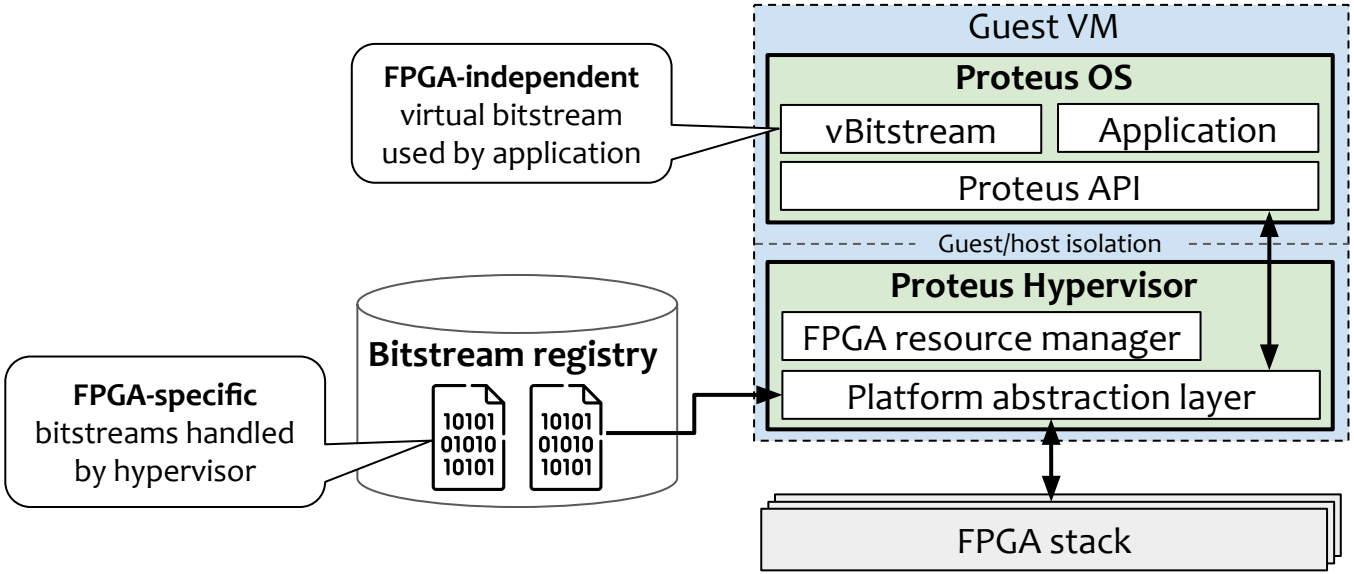
- ~~Motivation~~
- Overview
- Implementation
- Evaluation

Overview

Proteus abstracts the four key factors of FPGA heterogeneity in the cloud

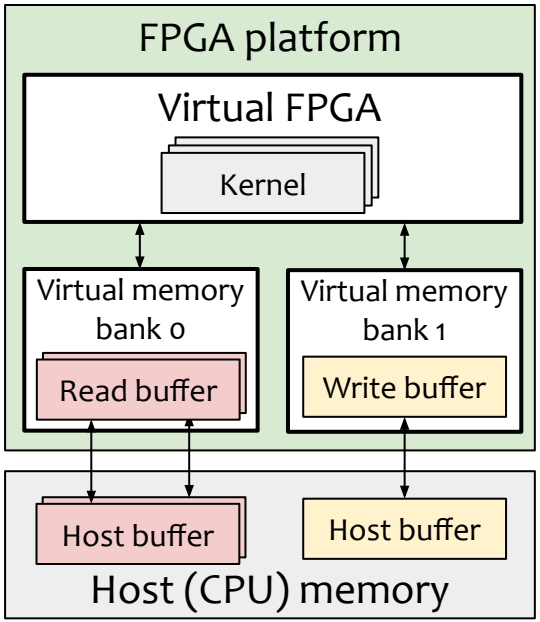


OS and Hypervisor



The Proteus virtualization decouples applications from heterogeneous FPGA stacks

Programming Model and API

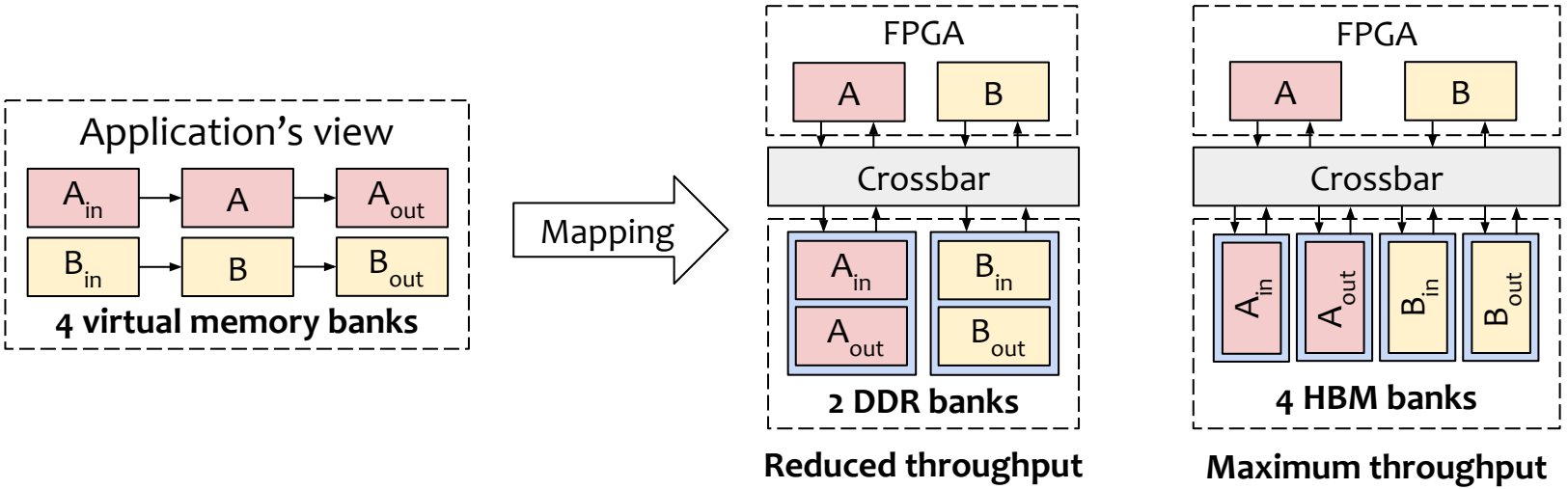



```
1  void vadd_host() {
2    /* Create kernels and buffers */
3    in1  = proteus::buffer(size, in1_host);
4    in2  = proteus::buffer(size, in2_host);
5    out  = proteus::buffer(size, out_host);
6    vadd = proteus::kernel("vadd", {in1, in2, out, size});
7
8    /* Allocate buffers and kernels to the FPGA platform */
9    platform = proteus::platform(virt_bs);
10   platform.add(vadd, in1, in2, out);
11
12   /* Execute the kernel */
13   platform.execute(vadd);
14   platform.sync(vadd);
15 }
```

The platform-agnostic Proteus API abstracts vendor-specific APIs

Memory Virtualization

Transparent virtualization and optimizations including bank assignment



Proteus' abstracts memory details and transparently performs optimizations

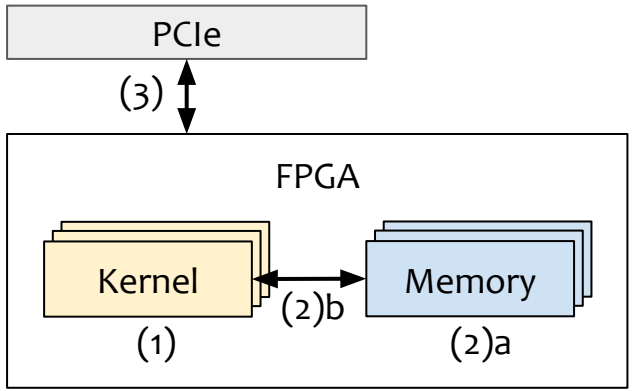
Performance-aware Scheduling

Performance analyzer generates performance metadata for the scheduler based on

(1) Kernel clock frequency & port widths

(2) Memory
  a) Type, bank configuration
  b) Throughput, access patterns

(3) PCIe bandwidth



Proteus' scheduler selects the best-performing FPGA for each task

Outline

- ~~Motivation~~
- ~~Overview~~
- Implementation
- Evaluation

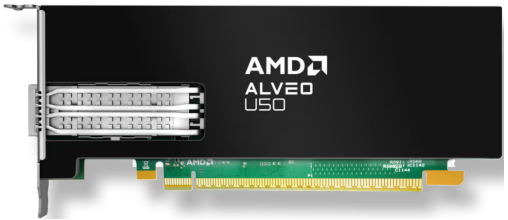
Implementation

- Proteus OS and API
  - based on includeOS
- Proteus Hypervisor
  - based on solo5
- Two supported vendor-specific platforms
  - AMD Vitis/XRT (U50 & U280)
  - Intel FPGA SDK for OpenCL (emulation)

includeOS

```
                |       ___|
  __|   _ \     |   _ \ __ \
\__ \ (   |   | | (   |   ) |
____/\___/  _|\___/____/
```

AMD XILINX



altera
An Intel Company

OpenCL™

Outline

- ~~Motivation~~
- ~~Overview~~
- ~~Implementation~~
- Evaluation

Evaluation

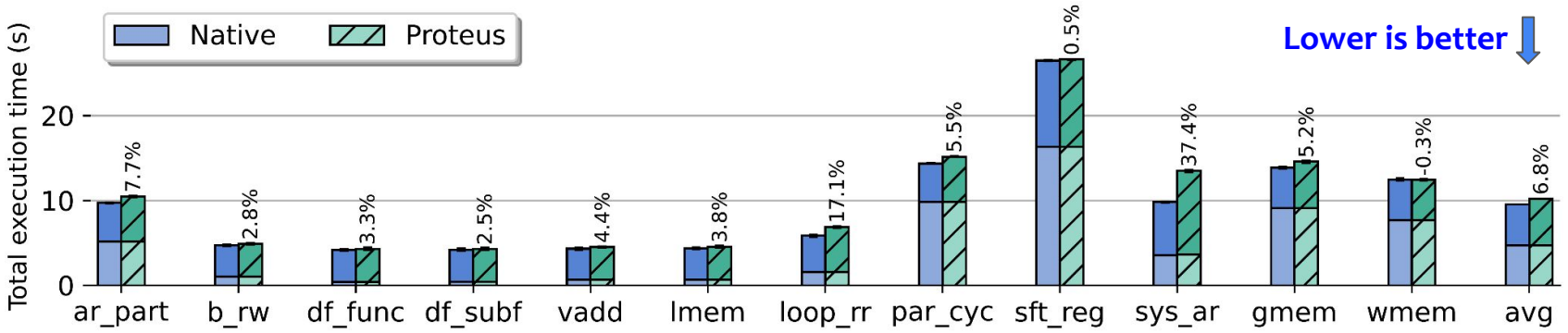
- Heterogeneous cluster
  - 1 leader node, 3 worker nodes with FPGA boards
- FPGA boards
  - AMD Alveo U50: 8GB HBM
  - AMD Alveo U280: 8GB HBM + 32GB DDR4
  - Emulated Intel Stratix 10: 1GB DDR4

Evaluation questions (more in the paper)

1. What is the **performance impact** of the Proteus virtualization stack?
2. How effective are Proteus' **memory optimization** techniques?
3. How does Proteus scale in a **multi-node cluster**?

Evaluation: Performance

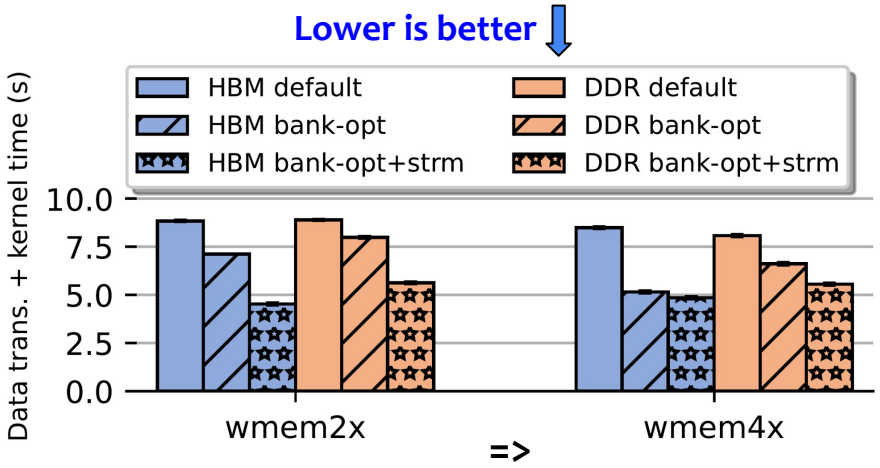
- Vitis Accel Examples on U50 against **native** (w/o Proteus)
- Lower bars: data transfer + kernel time



Proteus introduces an **average overhead of 6.8%** compared to native execution

Evaluation: Memory Virtualization

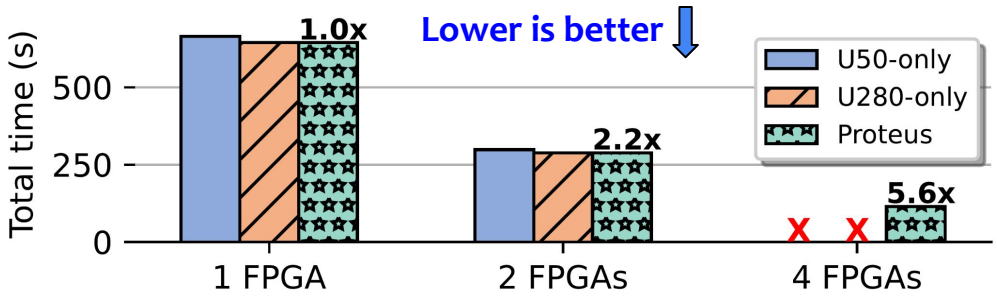
- **default**
  - kernels using single memory bank
- **bank-opt**
  - kernels using multiple memory banks
- **strm**
  - overlapping I/O & kernel execution



Proteus' memory optimizations can **speed up** execution time by up to **1.95x**

Evaluation: Scheduling

- Cluster with two U50 & two U280
- 16 deploy requests of same application



Proteus achieves up to a **5.6x speedup** with **4 FPGAs** **vs 1 FPGA**

Summary

Heterogeneous properties of FPGAs complicate development and deployment

- **Bitstream incompatibility**
- **Vendor-specific APIs**
- **Heterogeneous memory**
- **Performance gaps**

Proteus: Heterogeneous FPGA virtualization in the cloud

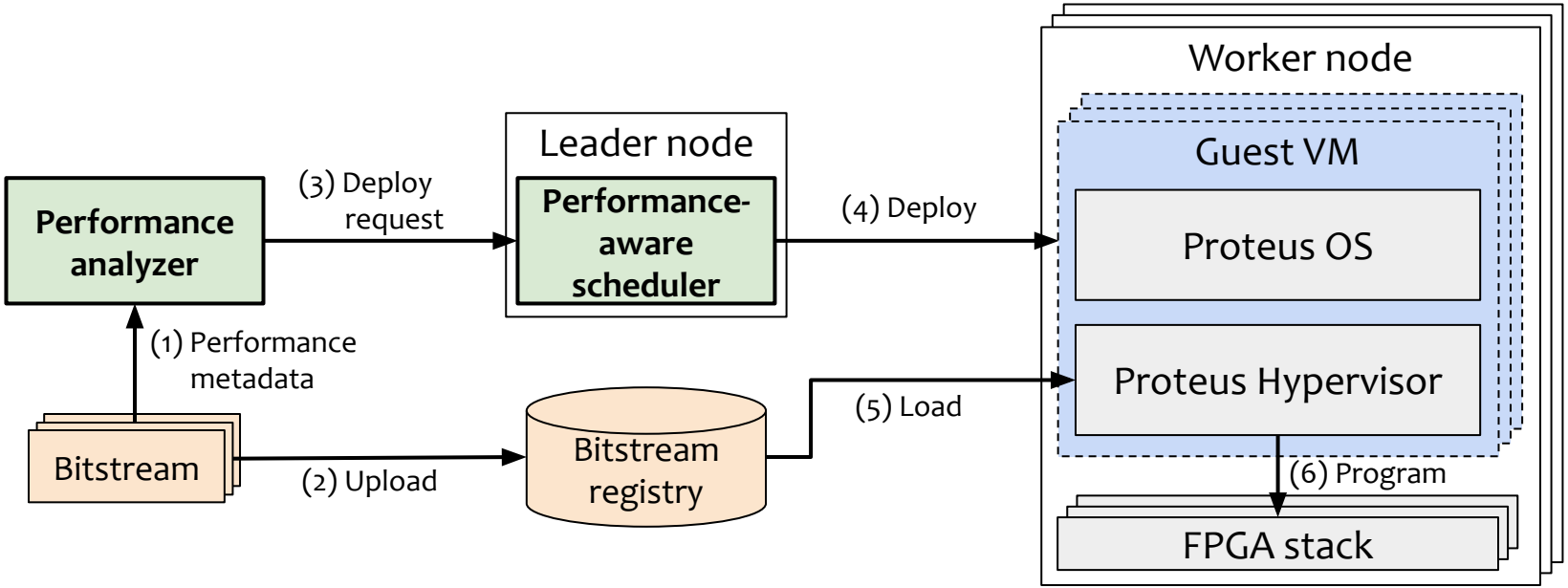
- **FPGA virtualization stack**
- **Unified, platform-agnostic API**
- **Memory virtualization**
- **Performance-aware scheduler**

Paper **Code**

github.com/TUM-DSE/proteus

Backup

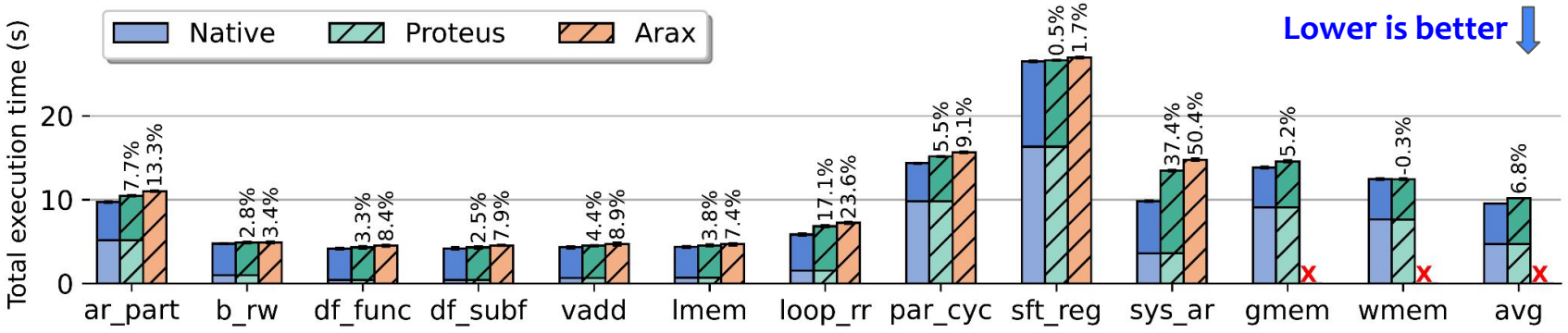
Performance-aware Scheduler



Proteus' scheduler selects the best-performing FPGA for each task

Evaluation: Performance

- Vitis Accel Examples on U50 against **native** (w/o Proteus) and **Arax [SoCC'22]**
- Lower bars: data transfer + kernel time



Proteus introduces an **average overhead of 6.8%** compared to native execution