HPCS

High Performance Computer System Lab

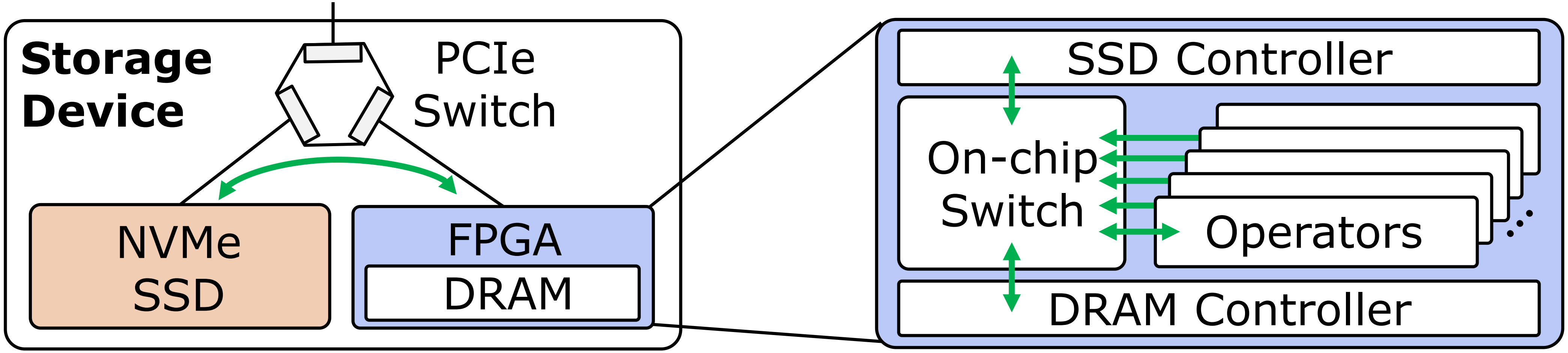
A Fast and Flexible Hardware-based Virtualization Mechanism for Computational Storage Devices

Dongup Kwon, Dongryeong Kim, Junehyuk Boo, Wonsik Lee, and Jangwoo Kim

Department of Electrical and Computer Engineering, Seoul National University

Background: Computational Storage

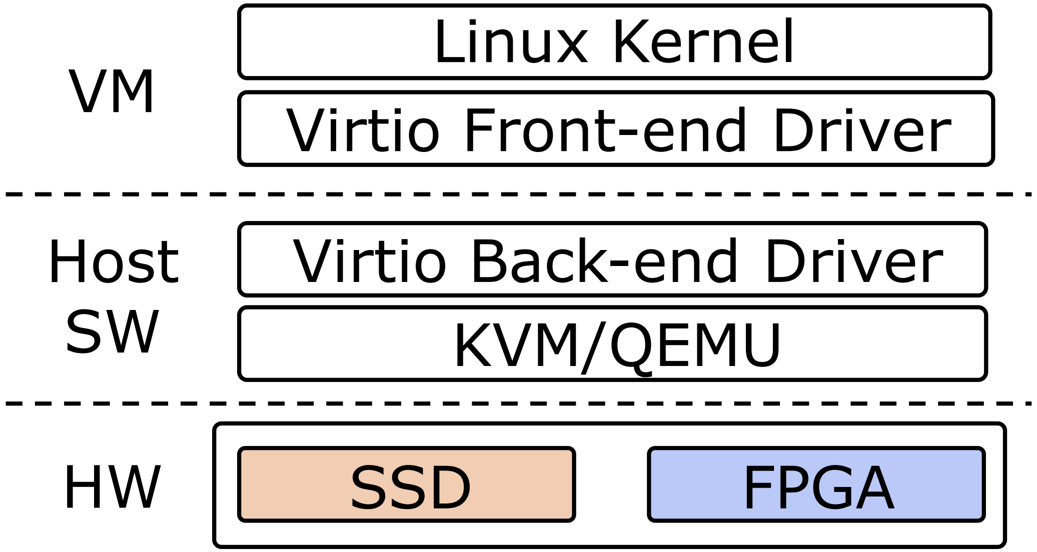
- **SSD-FPGA integration for near-storage processing**
  - Fast data transfers between the storage and computation units
  - Programmable operators and on-chip interconnects in an FPGA



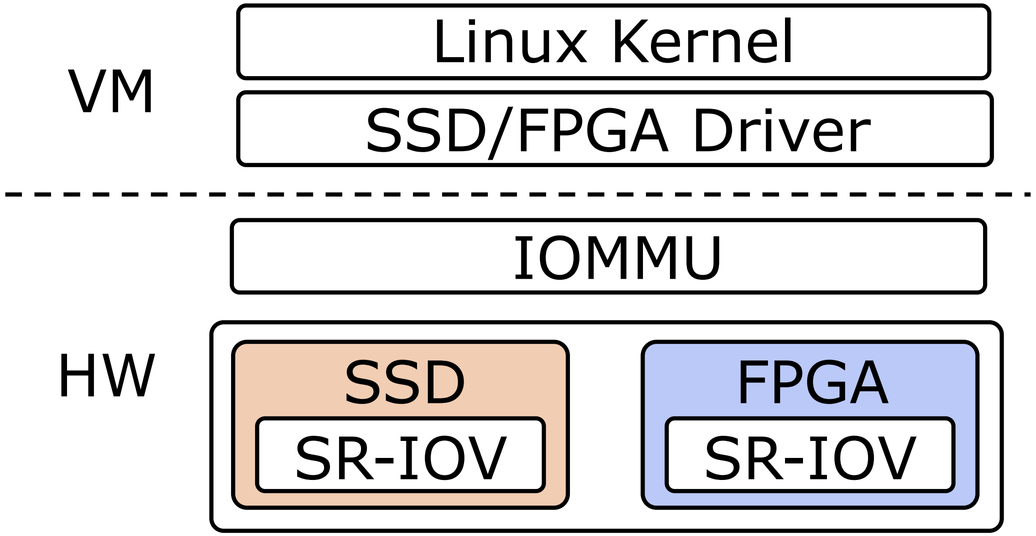
Computational storage = SSD + FPGA + near-storage processing

Background: I/O Virtualization

- **SW-based virtualization:** Paravirtualization (VirtIO)
- **HW-assisted virtualization:** Passthrough, SR-IOV, FVM*



Paravirtualization



SR-IOV

I/O virtualization enables resource sharing between VMs.

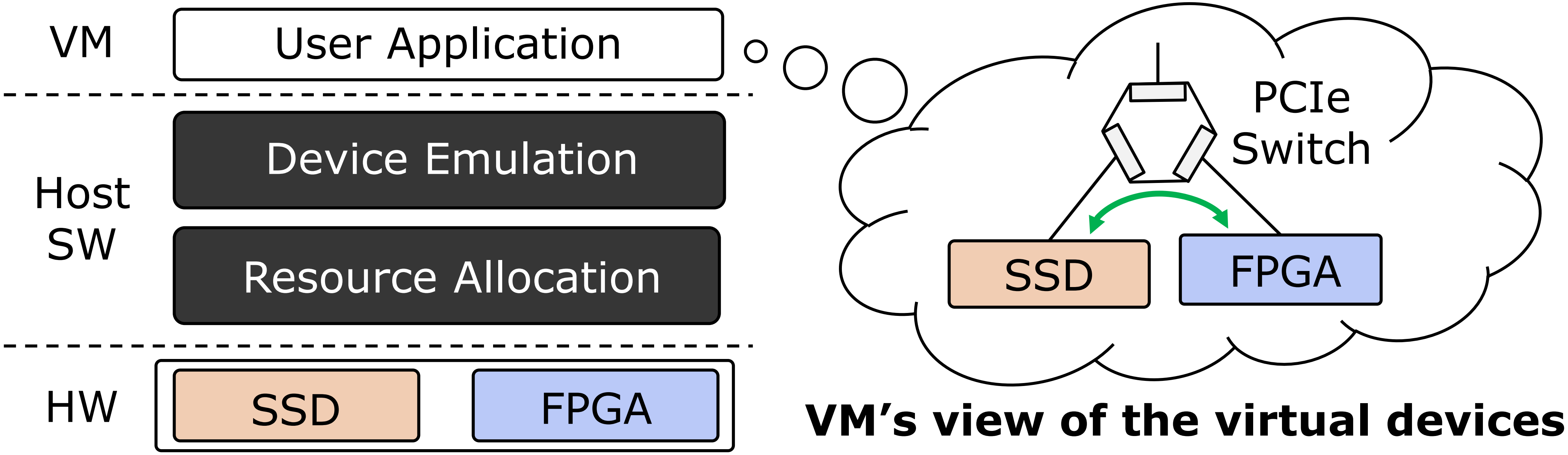
FVM: FPGA-assisted Virtual Device Emulation for Fast, Scalable, and Flexible Storage Virtualization, OSDI 2020

Outline

- Background
- **Motivation**
  - SW-based virtualization for computational storage
- FlexCSV: HW-assisted Virtualization Stack
- Evaluation
- Conclusion

SW-based Virtualization Approach

- SW emulation of SSD-FPGA integrated devices
- Host SW-level device resource allocation and scheduling

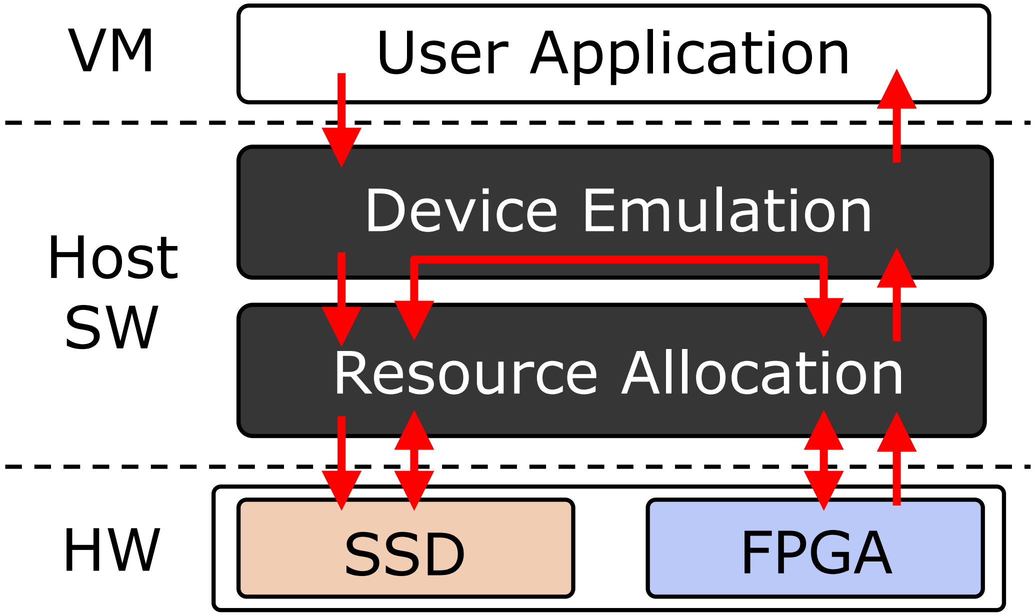


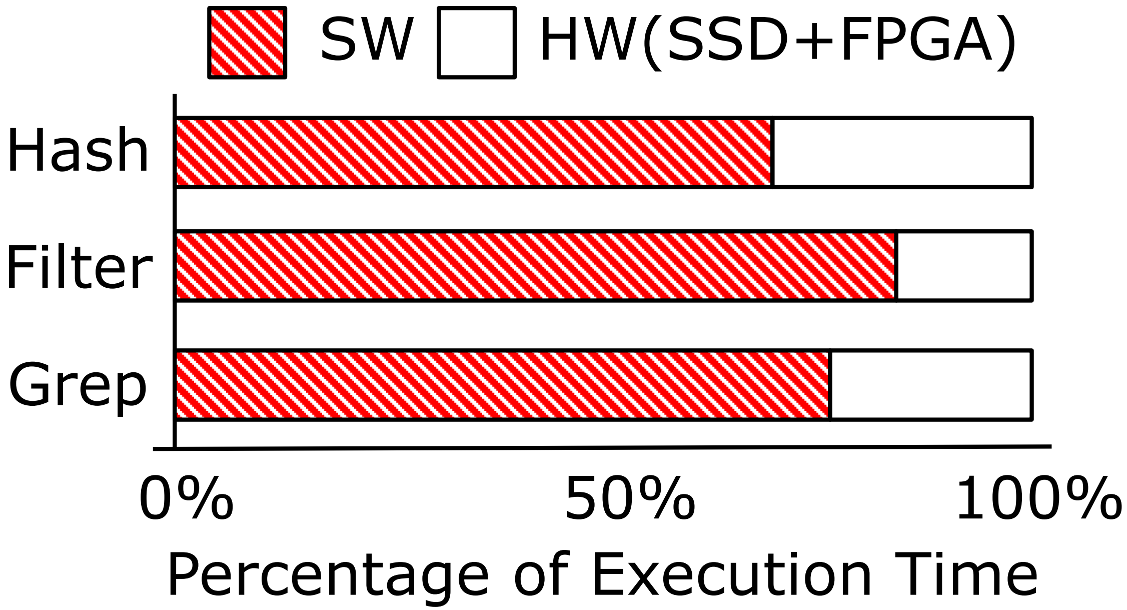
VM's view of the virtual devices

SW-based virtualization provides flexible virtual device construction mechanisms.

Limitation #1: CPU-centric Device Emulation

- **CPU-centric device orchestration & data transfers**
- **Cannot achieve full potential of near-storage processing**

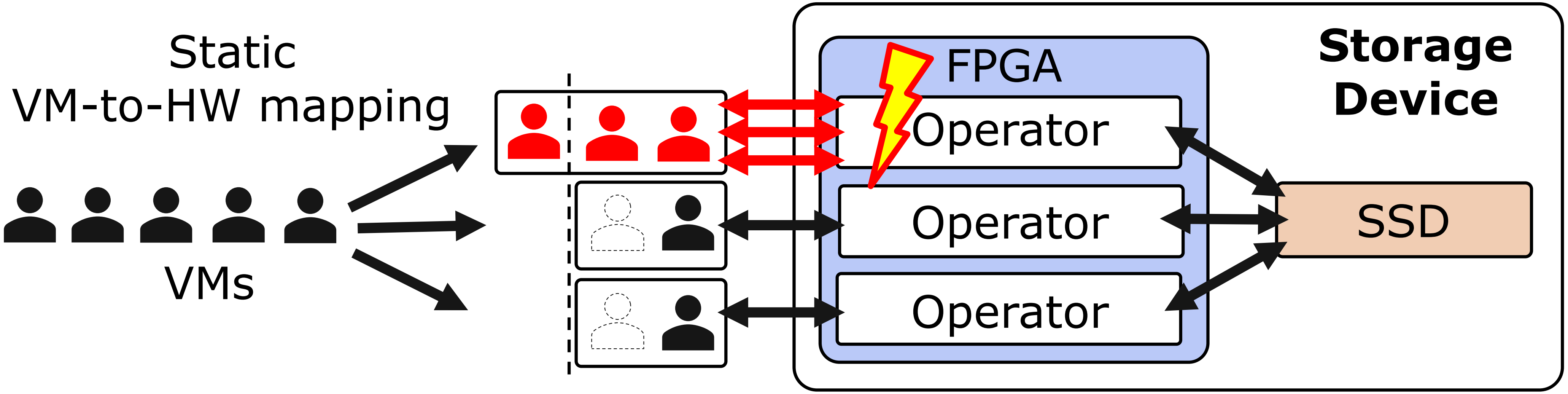




The bottleneck shifts to the SW components in a virtualized environment.

Limitation #2: Static Resource Allocation

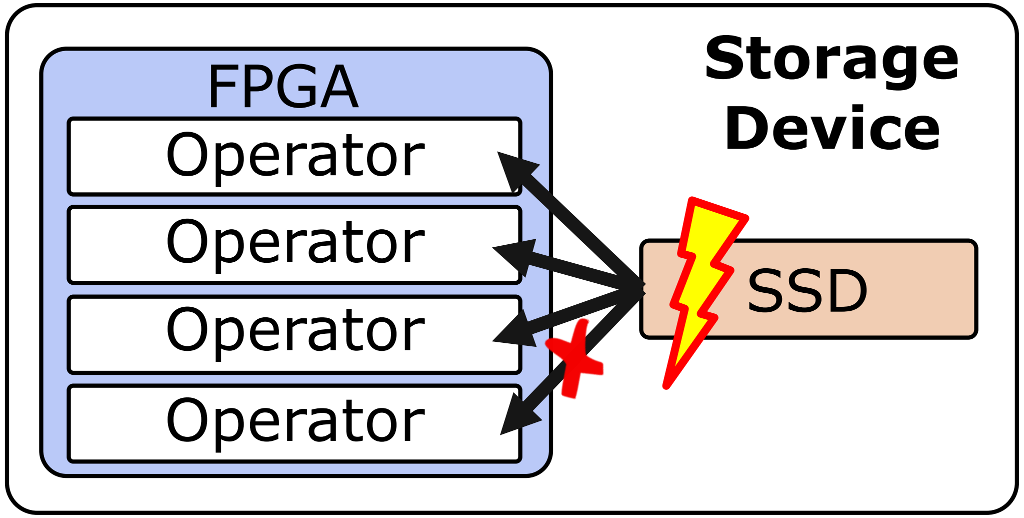
- **Static VM-to-HW resource allocation & scheduling**
- **Cannot achieve cost-effectiveness due to inefficient resource sharing**



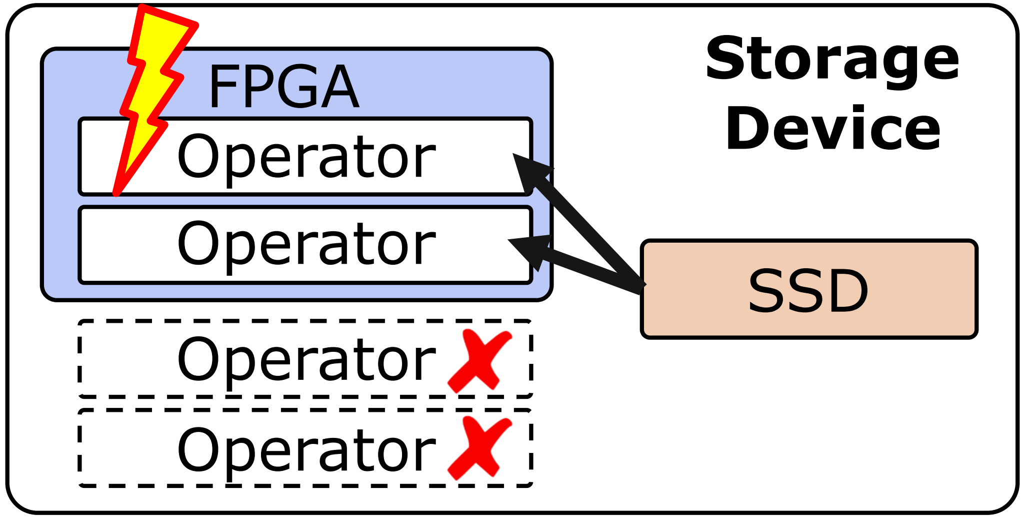
Static resource allocation incurs extra costs for the additional HW resources to meet QoS requirements.

Limitation #3: Coupled HW Architecture

- **SSD-FPGA coupled designs & fixed provisioning**
- **Cannot provide flexible device/resource configurations**



FPGA BW > SSD BW



FPGA capacity < SSD capacity

SSD-FPGA coupled architectures suffer from limited device scalability.

Design Goals

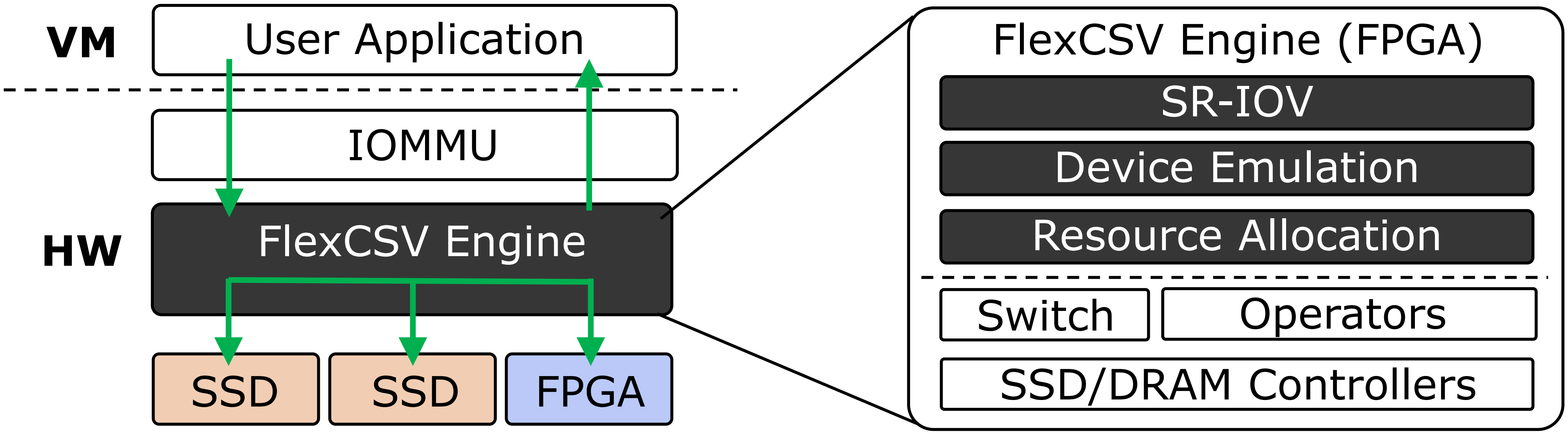
Design Goals	SW-based Virtualization	
Device Sharing	☑	*Trap-and-emulate*
High Performance	☒	*CPU-centric orchestration*
Low Cost	☒	*Static resource allocation*
Device Scalability	☒	*Tightly-coupled architecture*

Outline

- Background
- Motivation
- **FlexCSV: HW-assisted Virtualization Stack**
- Evaluation
- Conclusion

FlexCSV: SW/HW Architecture

- **HW virtualization for computational storage**



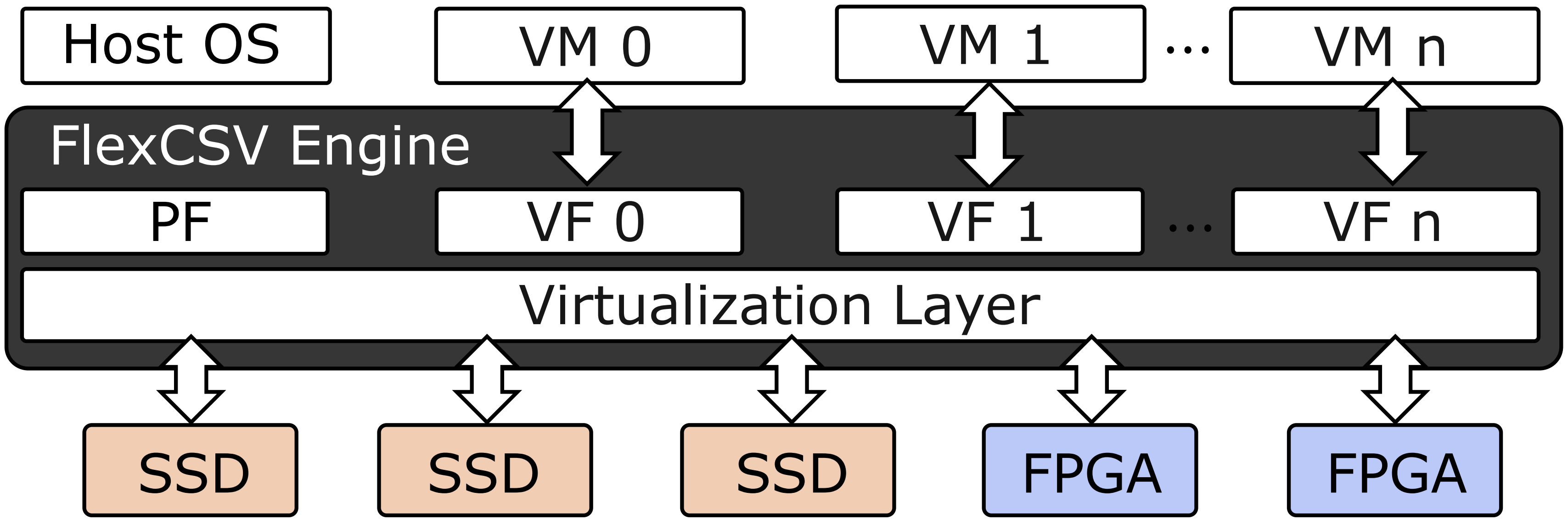
FlexCSV offloads a virtualization stack for computational storage devices.

FlexCSV: Key Ideas

Design Goals	FlexCSV		Key Ideas
Device Sharing	☑	→	*HW-assisted virtualization (including SR-IOV)*
High Performance	☑	→	*HW-level resource orchestration*
Low Cost	☑	→	*Dynamic resource allocation*
Device Scalability	☑	→	*SSD-FPGA decoupled architecture*

Key Idea #1: HW-assisted Virtualization

- SR-IOV implementation in FlexCSV Engine
- SSD/FPGA sharing between VMs with direct HW access



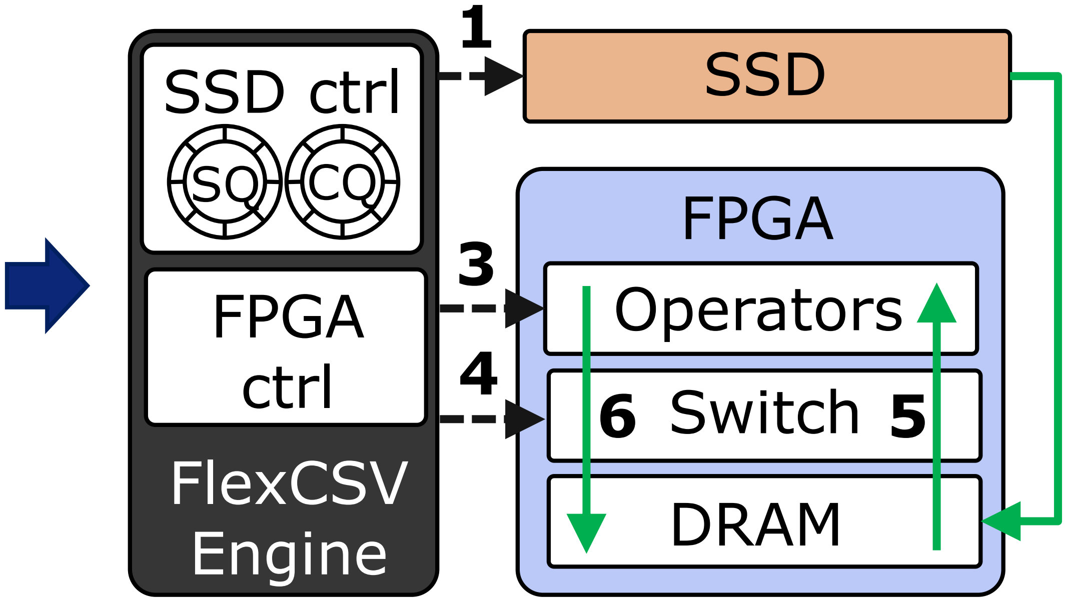
FlexCSV Engine virtualizes itself through SR-IOV and offers device sharing.

Key Idea #2: HW-level Orchestration

- **NVMe extension for data processing requests**
- **Guest/host OS bypassing and direct data communications**

```
struct nvme {
       u8   rw;
       u64  rsvd;
       u64  prp1;
       u64  prp2;
       u64  slba;
}
```

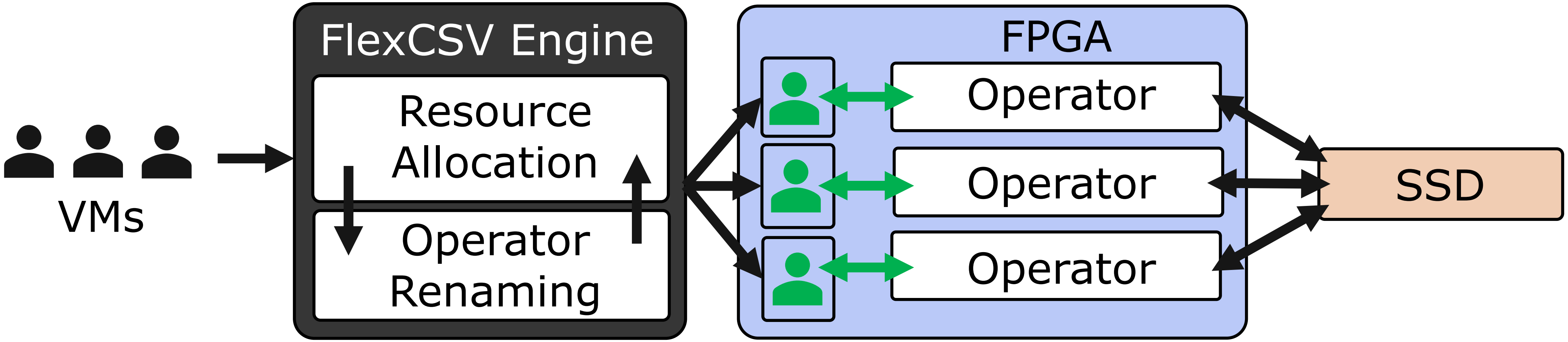
```
struct proc {
       u64  op;
       u64  src;
       u64  dst;
       u32  size;
       u64  param;
}
```



FlexCSV Engine orchestrates SSD and FPGA operations without SW arbitration.

Key Idea #3: Dynamic HW Allocation

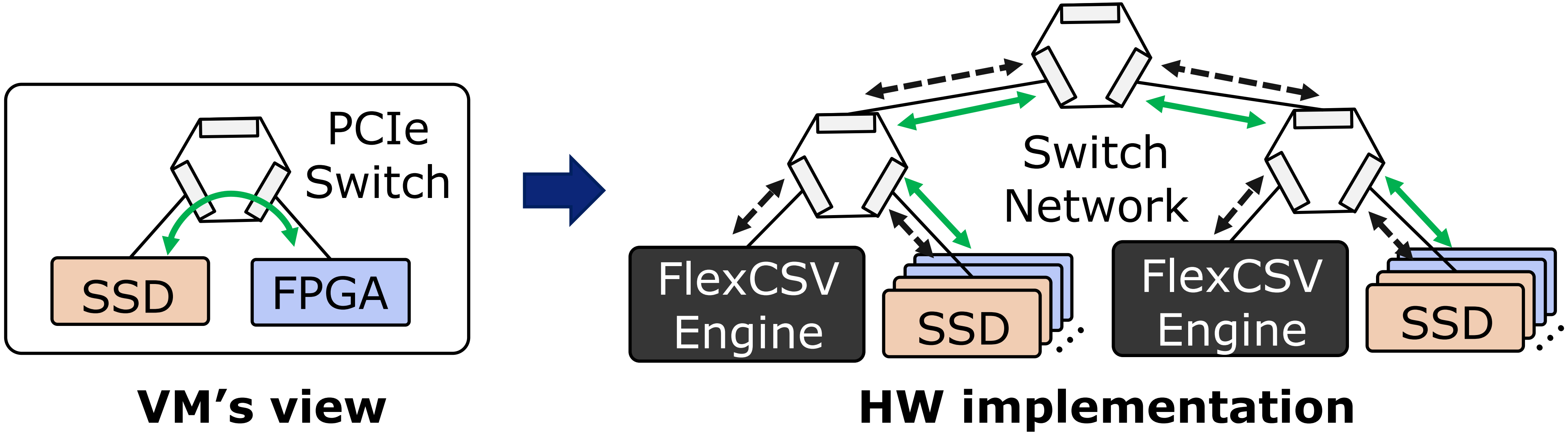
- Renaming of user-requested HW resources
- Efficient use of HW resources – High HW utilization



FlexCSV Engine implements HW renaming logic for dynamic resource allocation.

Key Idea #4: SSD-FPGA Decoupled Architecture

- **Decoupled HW through board-level PCIe switches**
- **Scalable virtual devices with many PCIe-attached cards**



FlexCSV Engine provides scalable and flexible device/resource configurations.

FlexCSV Prototype

- **HW Prototype**
  - Supermicro Server 4029GP-TRT2
  - Intel Optane 900P SSDs
  - Xilinx U250 FPGA (FlexCSV Engine)
- **SW Frameworks**
  - Ubuntu / Linux kernel v5.3
  - KVM / QEMU v3.0

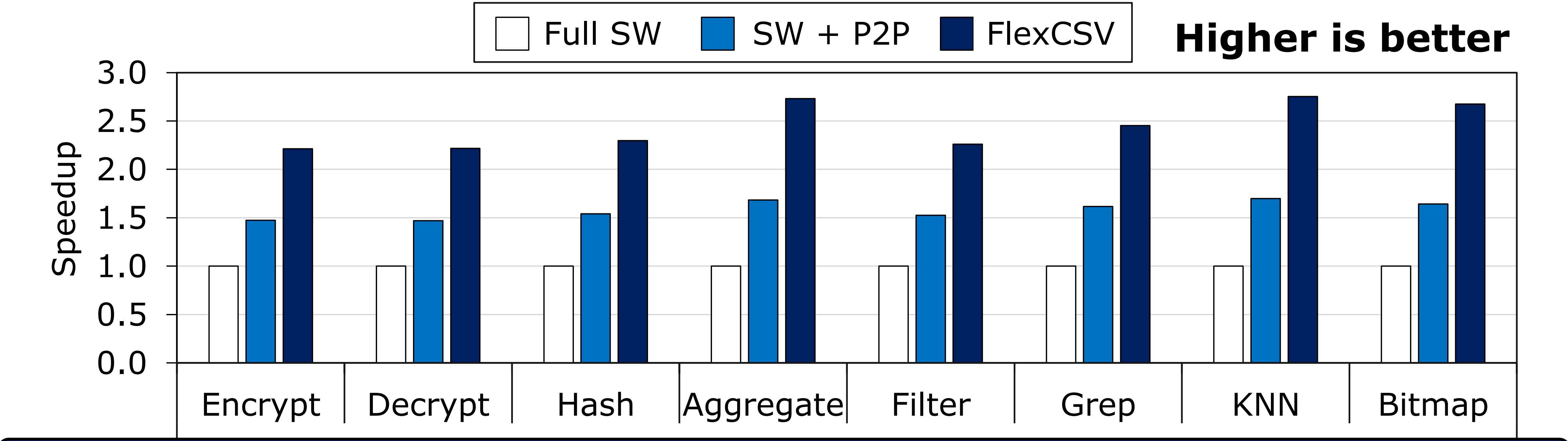
FlexCSV prototype is built on off-the-shelf HW devices and open-source SW.

Outline

- Background
- Motivation
- FlexCSV: HW-assisted Virtualization Stack
- **Evaluation**
- Conclusion

Near-storage Processing Performance

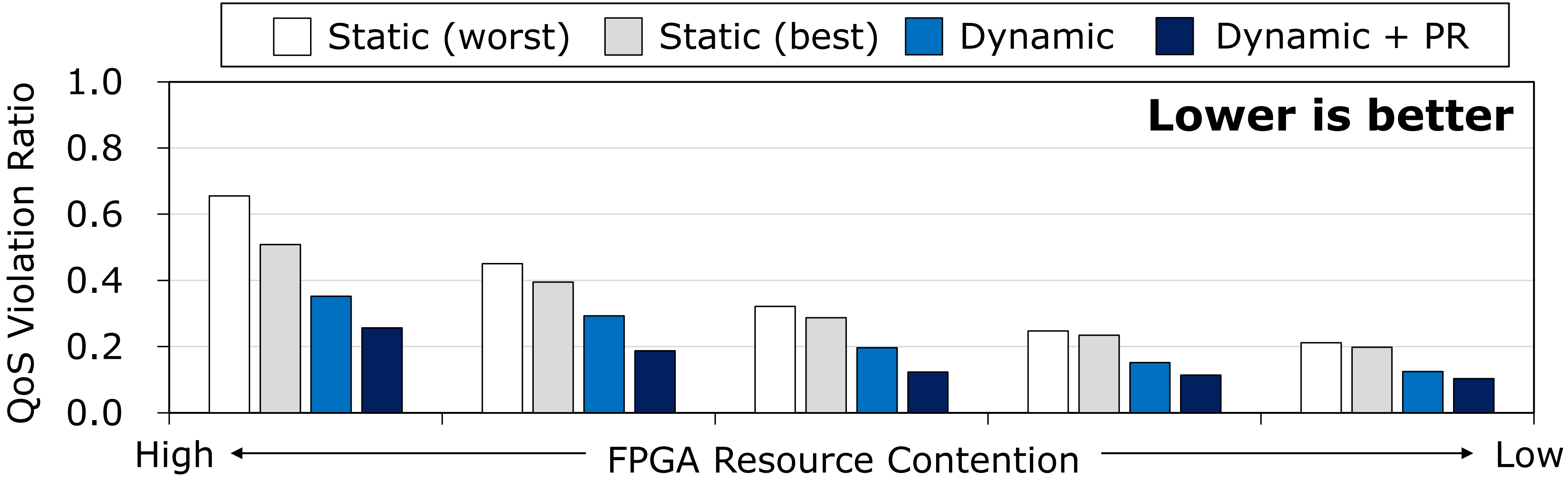
- **8 FPGA benchmarks with direct SSD read & write**
- **Guest/host OS bypassing + fast data copy → 2.4x speedup**



FlexCSV achieves high performance through its HW-assisted virtualization.

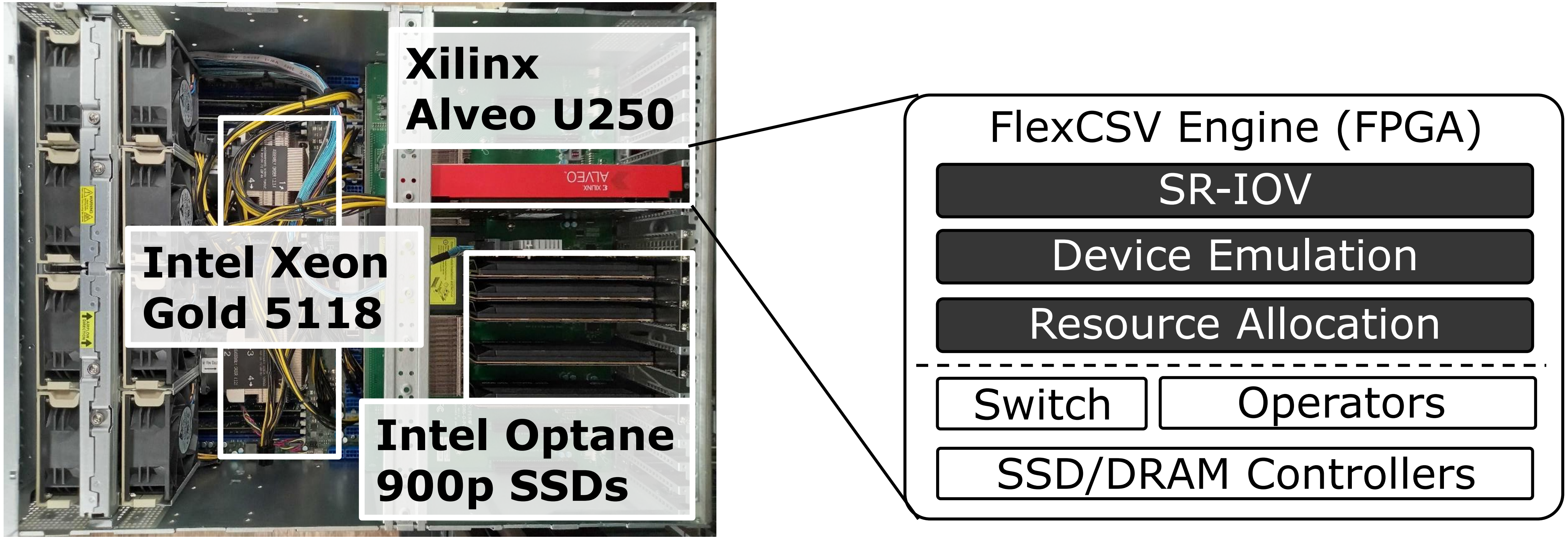
QoS Evaluation with Oversubscription

- 2 operators + 4 VMs with different request rates
- Dynamic allocation + partial reconfiguration → 2.4x better



FlexCSV achieves lower QoS violations through its efficient HW resource use.

Thank You!



FlexCSV Engine (FPGA)

SR-IOV

Device Emulation

Resource Allocation

Switch

Operators

SSD/DRAM Controllers

A Fast and Flexible Hardware-based Virtualization Mechanism for Computational Storage Devices, ATC 2021

Dongup Kwon, dongup@snu.ac.kr, https://hpcs.snu.ac.kr/~dongup/