MDev-NVMe: A NVMe Storage Virtualization Solution with Mediated Pass-Through

Bo Peng[1,2], Haozhong Zhang[2], Jianguo Yao[1],

Yaozu Dong[2], Yu Xu[1], Haibing Guan[1]

[1] Shanghai Key Laboratory of Scalable Computing and Systems, *Shanghai Jiao Tong University;*

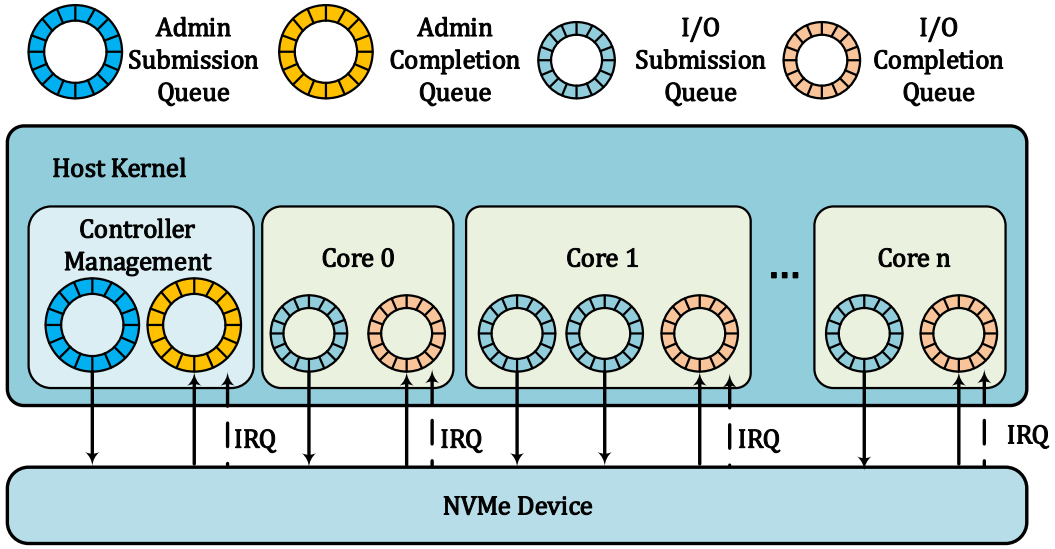
[2]*Intel Corporation;*

Outline

- Background
- Motivation
- Design
- Implementation
- Experiments
- Discussion
- Conclusion

Background

- NVMe Protocol
  - Design a logical device interface via a PCIe bus;
  - Deliver I/O operations with high throughput and low latency;
  - Admin or I/O Commands and Queues, Doorbells;
  - Submission Queue (SQ) and Completion Queue (CQ);
- Usage of NVMe SSDs
  - Storage for data-intensive datacenters;
  - accelerate I/O between system storage and other PCIe devices, such as GPUs; (BERGMAN, S., Spin, ATC'17)



Background

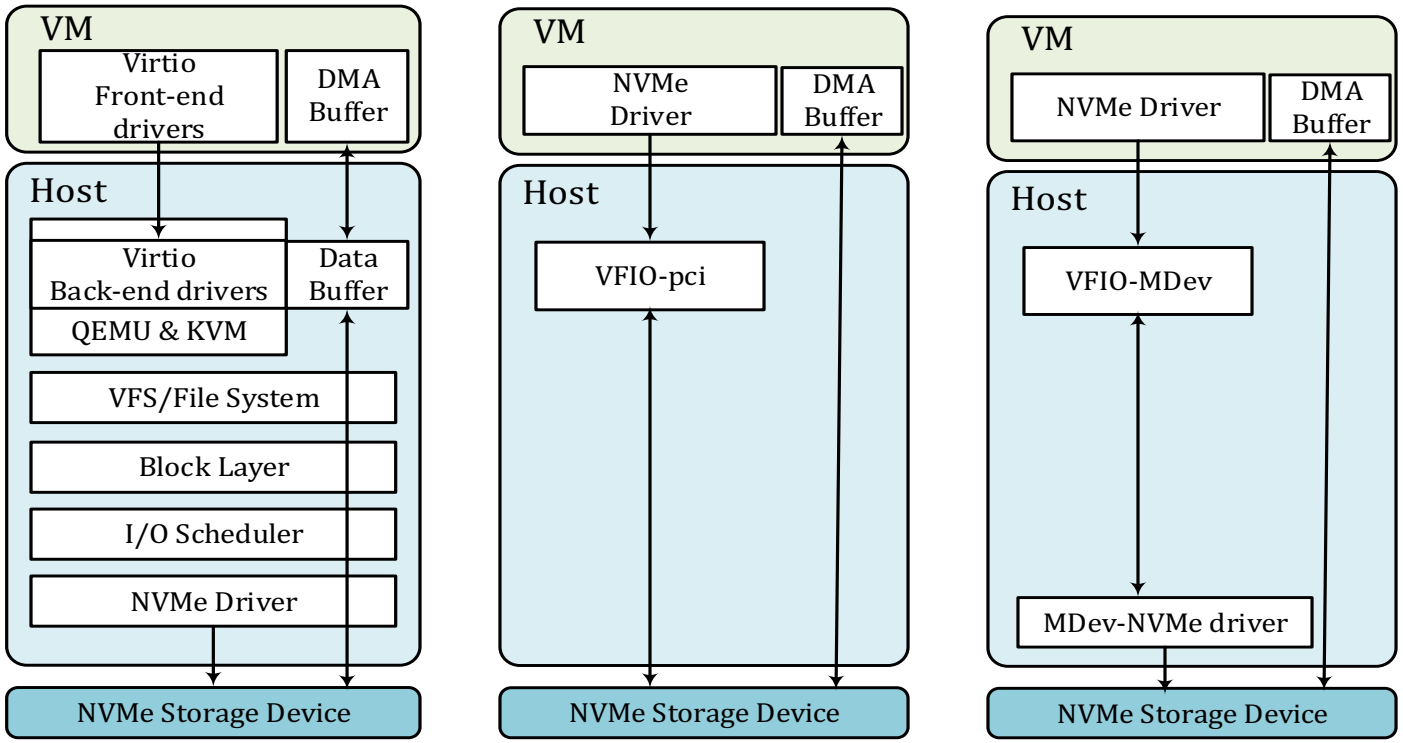
- I/O Virtualization
  - Optimize utilization of physical storage resource
  - Simplify storage management
  - Provide a simple and consistent interface
- NVMe Virtualization
  - Blind Mode
  - Virtual Mode
  - Physical Mode
- NVMe Mainstream Virtualization
  - VirtIO, Userspace driver in QEMU (Fam zheng), SPDK

Motivation

- Requirement of High-performance Virtualization for NVMe
  - High Throughput
    - Quick data access and parallel processing
  - Low Latency
    - Virtualization brings latency overhead when context switches happen
  - Device Sharing
    - Separate VMs on one device with good scalability
  - Device Feature
    - Guest uses original device drivers.
  - Migration Support
    - live migration relies on hypervisor management

Motivation

Different NVMe Comparison



VirtIO VFIO MDev-NVMe

- *Virtio* suffers readily apparent overhead from software layers.
- *VFIO* can not meet the requirements of device sharing.
- MDev-NVMe module manage all the statuses of queues with "VFIO-MDev" interface.

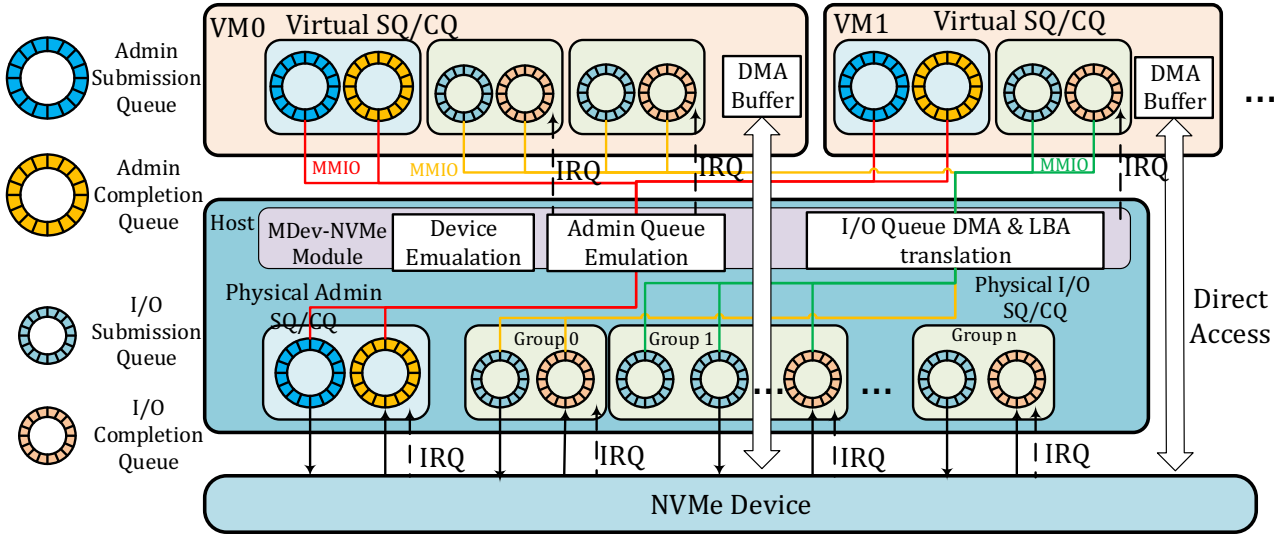
Design

- Mediated Pass-through (MPT)
  - An intermediate between direct pass-through and full emulated virtual device
  - Pass through performance-critical resource but traps and emulates privileged operations and resource
- Famous MPT research (MPT on GPU)
  - Kun Tian, et al: A Full GPU Virtualization Solution with Mediated Pass-Through. (ATC' 2014)
  - Yaozu Dong, et al :Boosting GPU Virtualization Performance with Hybrid Shadow Page Tables. (ATC' 2015)
  - Mochi Xue, et al: gScale: Scaling up GPU Virtualization with Dynamic Sharing of Graphics Memory Space. (ATC' 2016)

Design

Architecture

- A Full Virtualization
- MDev-NVMe Module
- Native Drivers in Guests

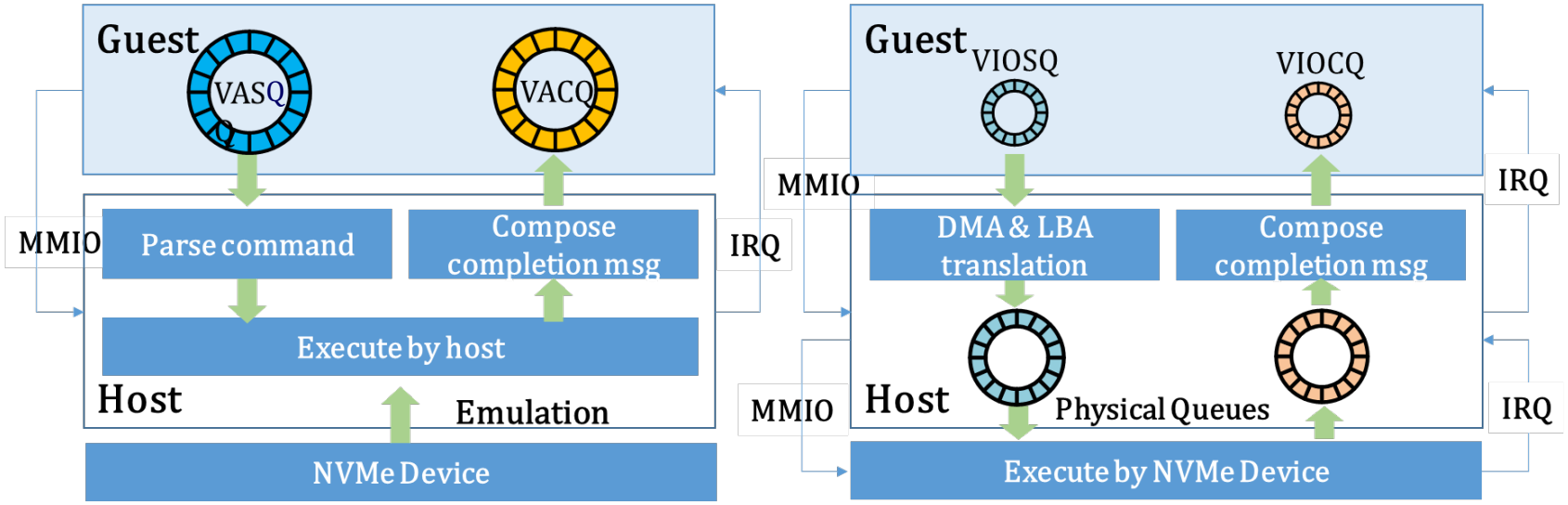


Technical Points:

- Device Emulation for PCIe
  - PCI registers & BAR
  - NVMe registers & logics
  - IRQ: INTx, MSI, MSI-X
- Admin Queue Emulation
  - Resource emulation
- I/O Queue Shadow
  - Resource pass-through
- DMA Buffer Access

Implementation

- Queue Handling
  - Emulation for Admin Queues
  - Queue Shadow from guest to host queues



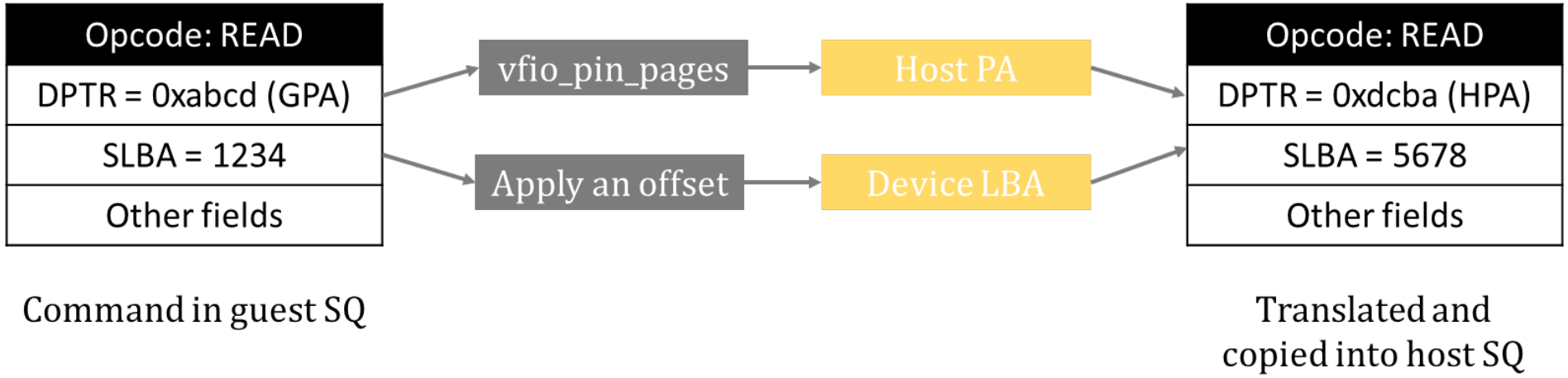
Admin Queue Emulation

I/O Queue Shadowing

Implementation

DMA & LBA translation

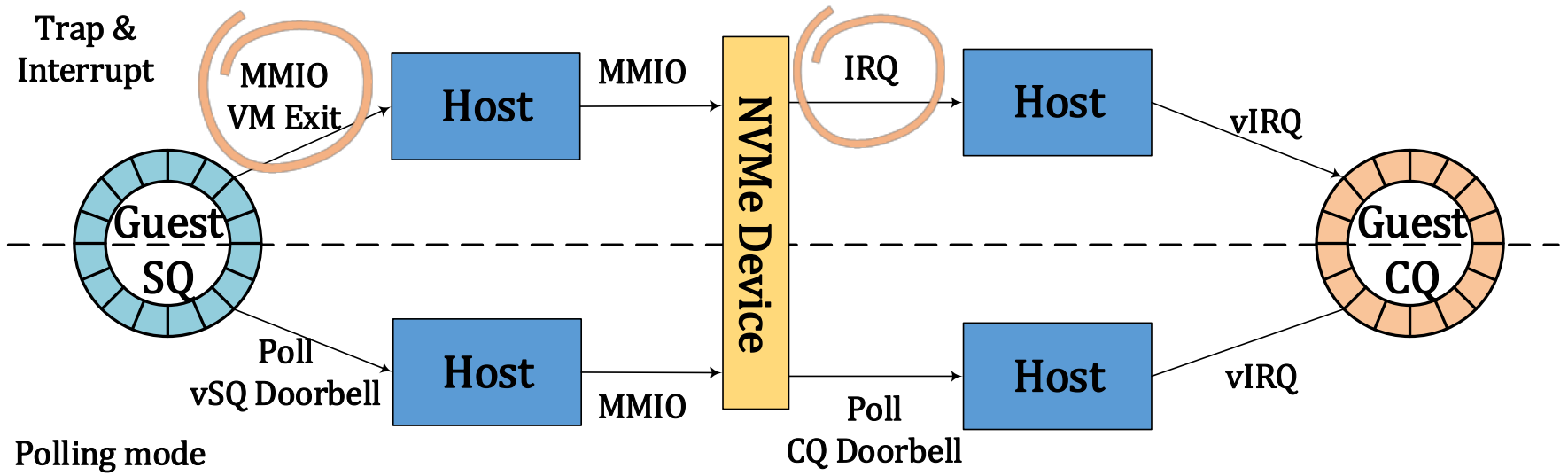
- Data pointer (DPTR) in IO commands points to a DMA buffer
  - The address is Guest Physical Address (GPA)
  - *vfio_pin_pages* translates GPA to HPA and pin the DMA buffer in host memory
- Start LBA is specified by SLBA in IO commands



implementation

2-Way Polling Mode

- Performance of non-polling not good enough
  - Each IO transaction requires an MMIO write to the doorbell and VM exit
  - IRQ becomes a big delay when IO is intensive

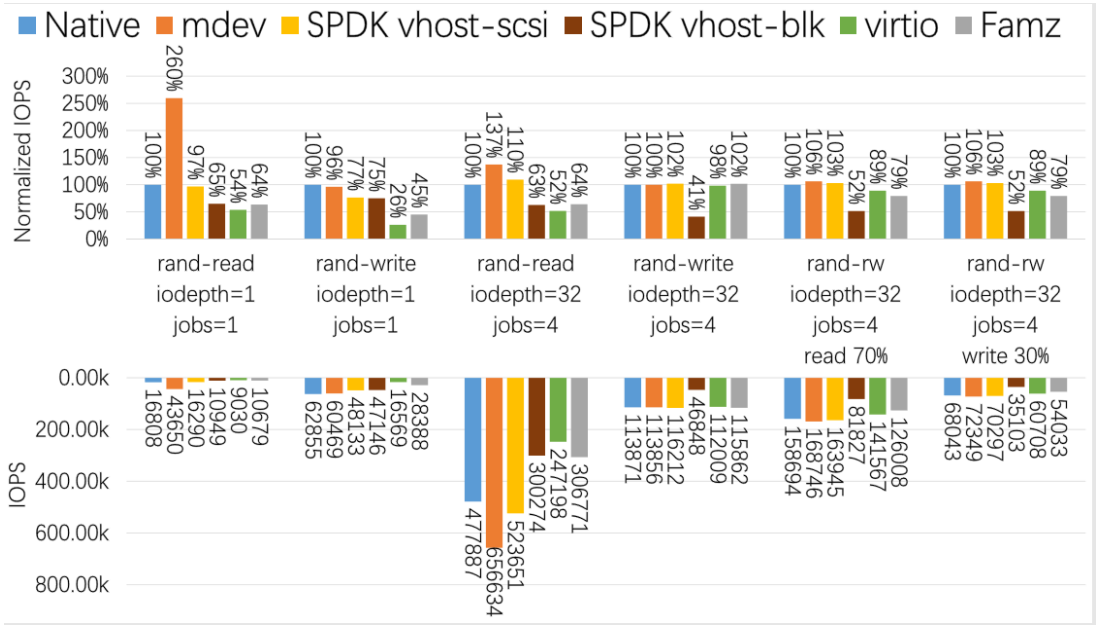


Experiments

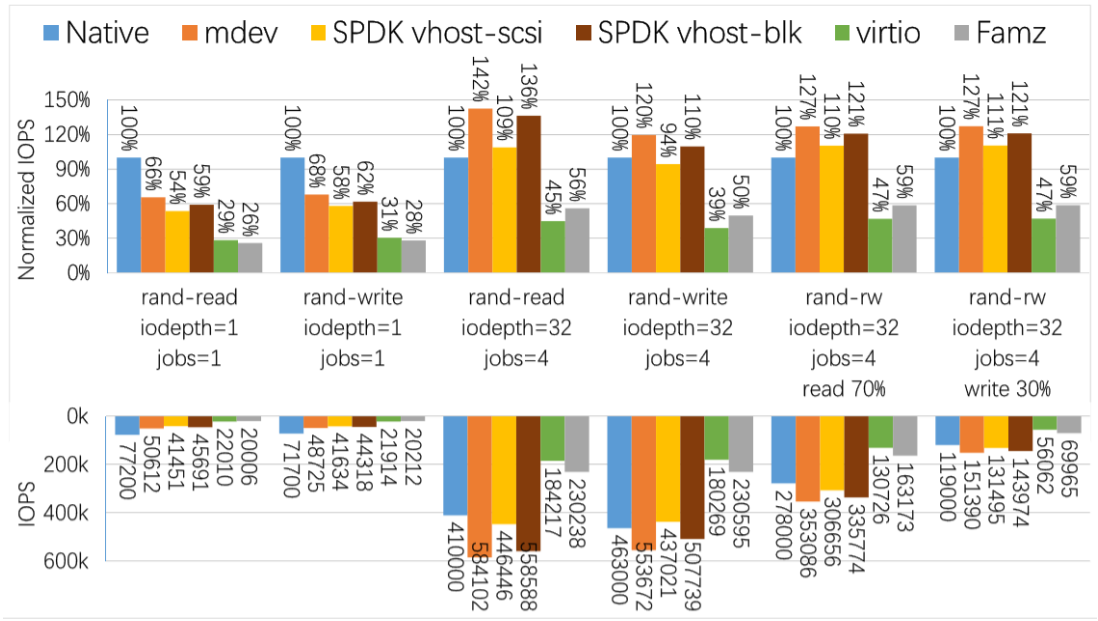
- Hardware Configuration
  - A server with 2 Intel Xeon CPU E5-2699 v3, 64GB system memory
  - Intel OPTANE DC P4800X SSD (375G)
  - INTEL SSD DC P3600 (400G)
- Application benchmark
  - Flexible I/O tester (FIO), 4K random read or write.
  - “libaio” as the fio engine, “O_DIRECT” for Direct I/O experiments
- Comparison Virtualization mechanism
  - Virtio, Famz Userspace Driver in QEMU, SPDK (vhost-scsi, vhost-blk)

Experiments

Throughput (IOPS)



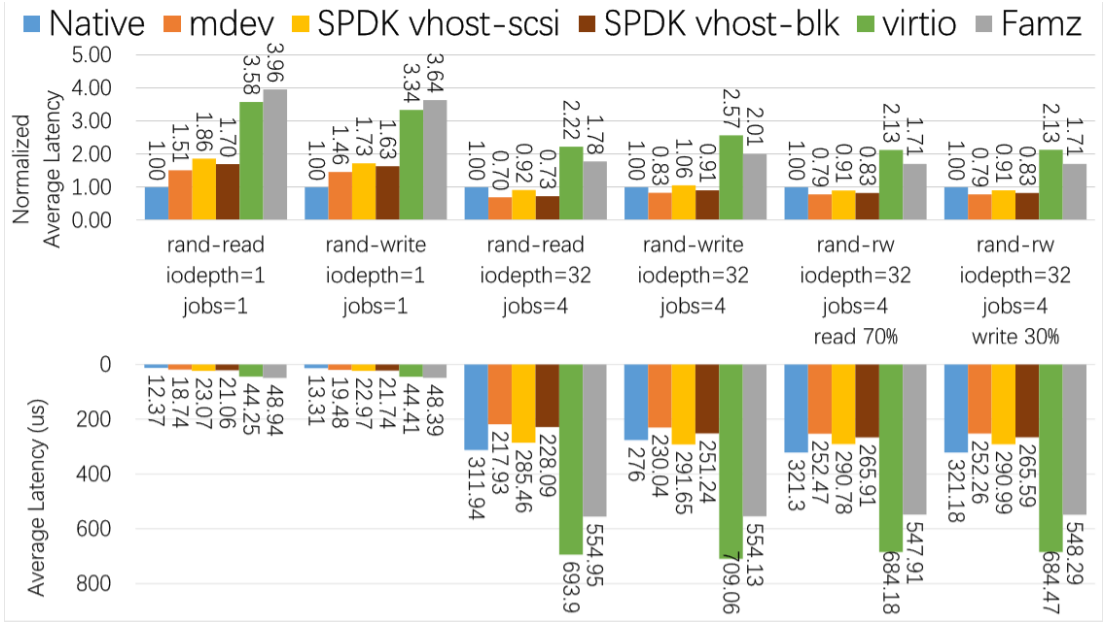
Optane P4800



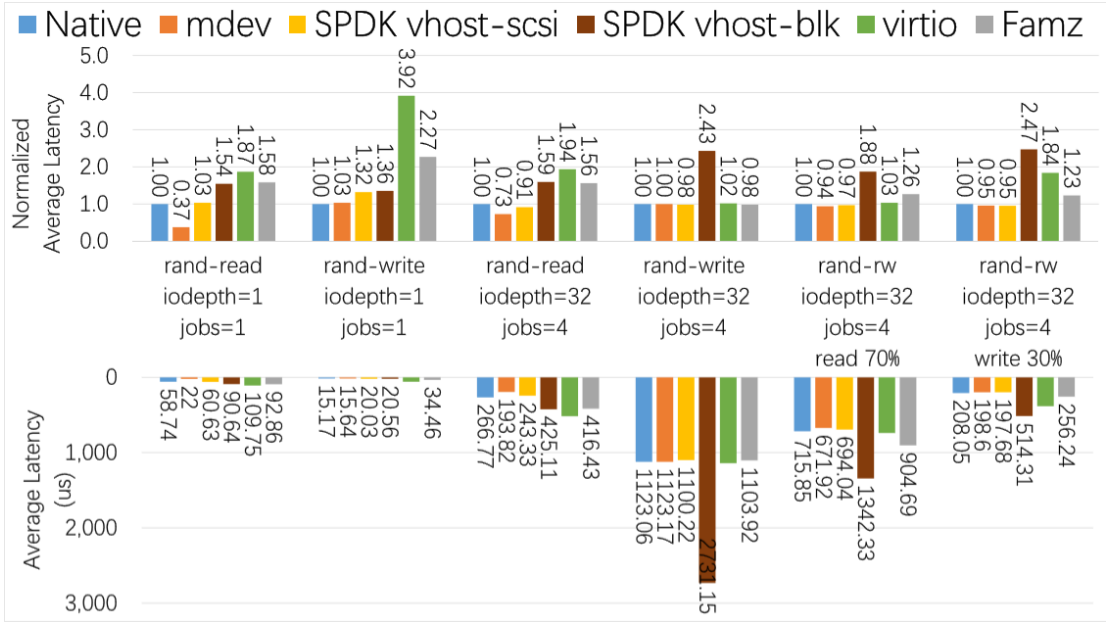
P3600

Experiments

Average Latency



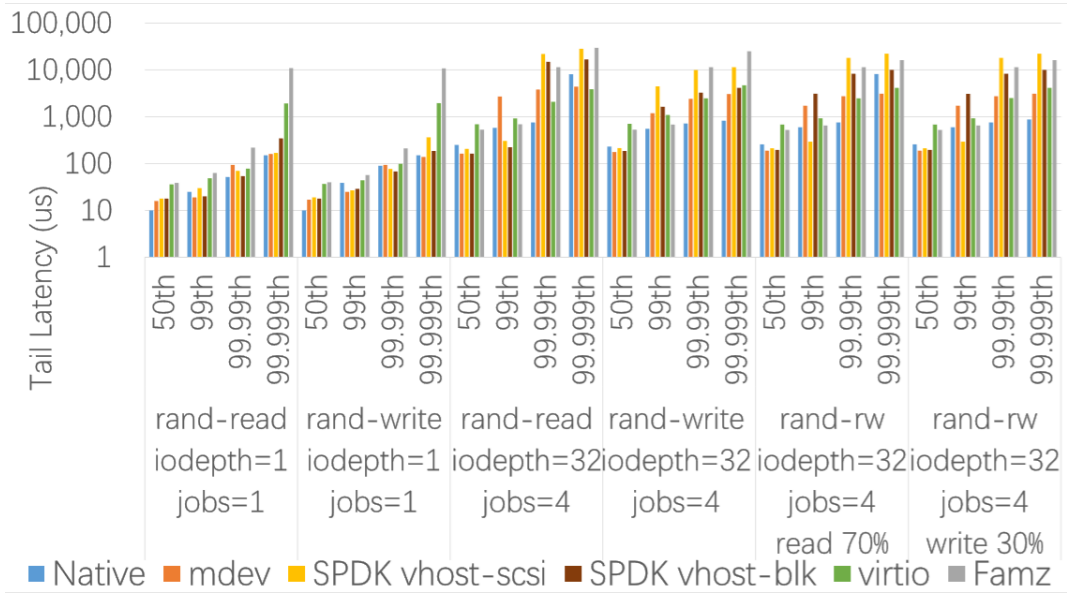
Optane P4800



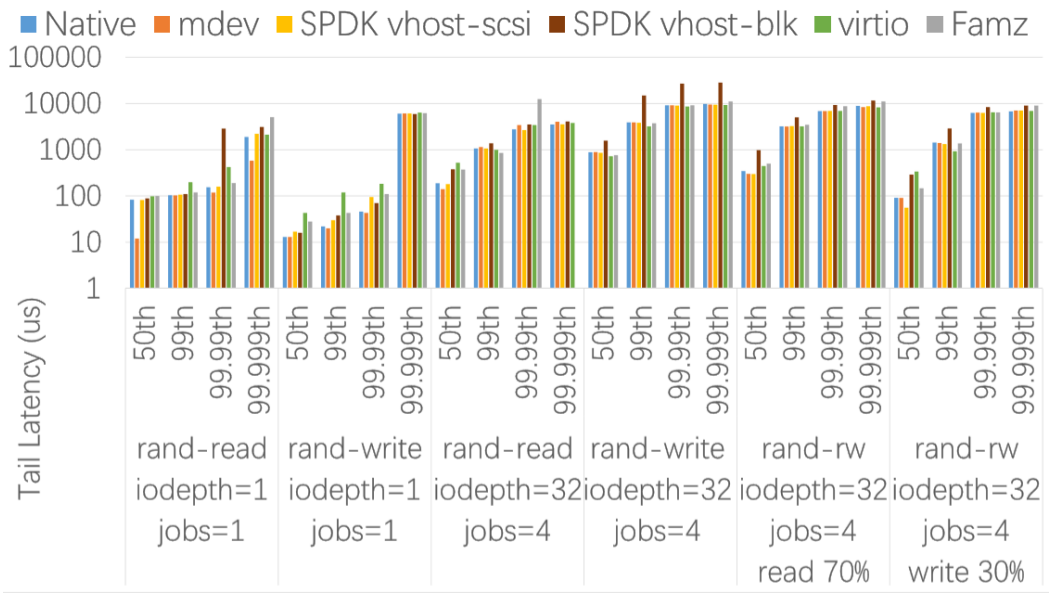
P3600

Experiments

QoS



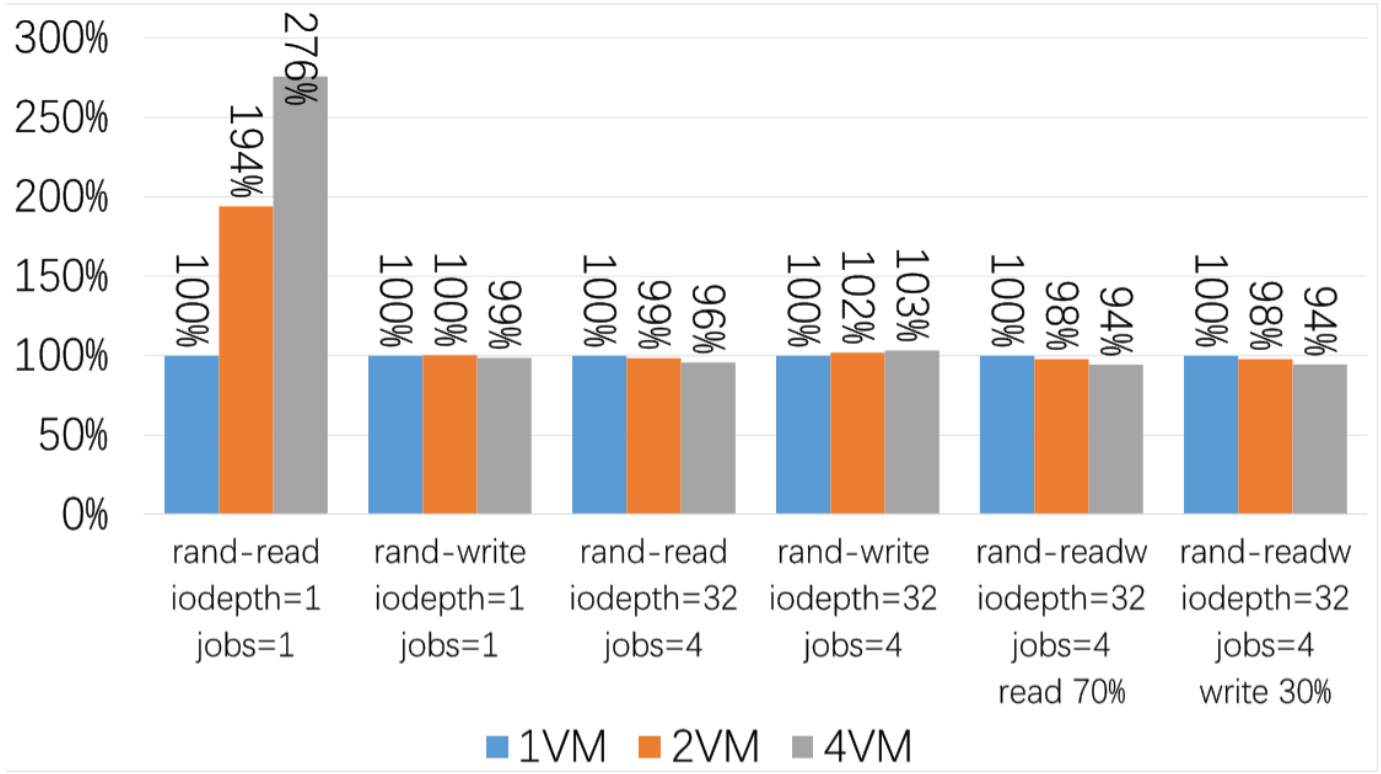
Optane P4800



P3600

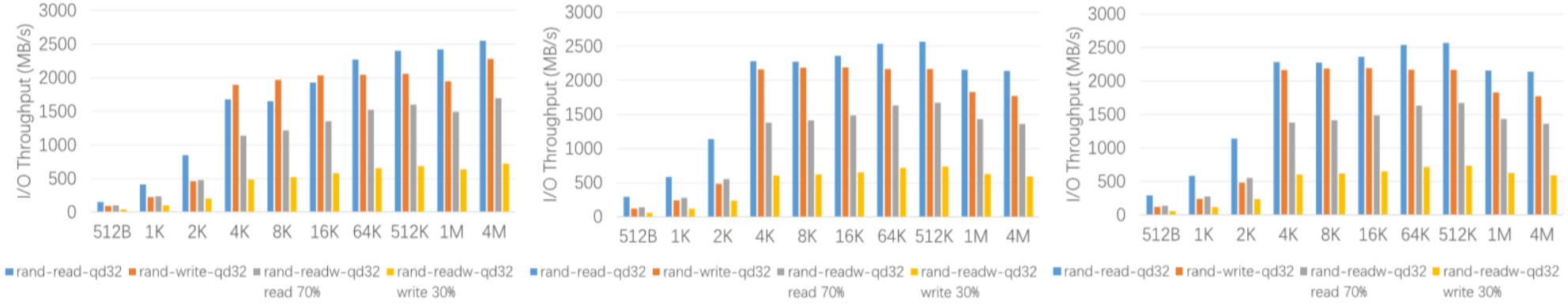
Experiments

- Device Sharing with good scalability



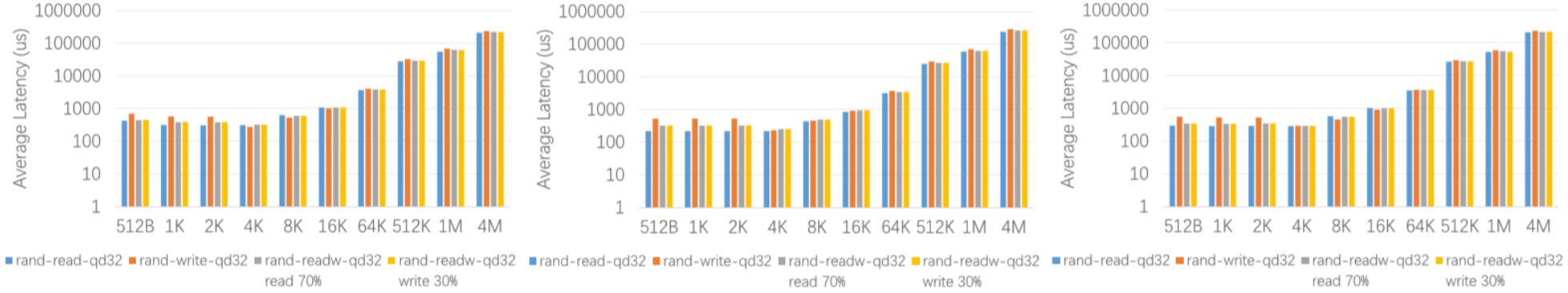
Experiments

Influence of I/O Blocksize



(a) Native performance (b) MDev-NVMe performance (c) SPDK performance

Optane P4800



(a) Native performance (b) MDev-NVMe performance (c) SPDK performance

P3600

Discussion

- Importance of polling (polling VS. interrupts)
  - The main agreement of MDev-NVMe and SPDK is taking active polling
  - The trap of the “vm-exit” needs additional context switches
  - polling is necessary when the I/O workloads are aggressive
  - To support adaptive polling with a mild policy for I/O acceleration

- We expect that high-performance I/O device will be designed with components to actively support or cooperate with the polling.

Conclusion

- MDev-NVMe is a full NVMe storage virtualization mechanism which takes a mediated pass-through as the main implementation.
- MDev-NVMe presents Admin queue emulations and I/O queues shadowing, achieving high throughput, low latency performance and a reliable scalability with device sharing feature.
- We offer large numbers of Fio benchmarks to provides evidence for MDev-NVMe, and give research conclusion about the all mainstream NVMe virtualization mechnisms.
- We raise focus and consideration into the influence of blocksize.
- We raise a discussion about polling vs. interrupts.

Thank you， Questions?