ACCELERATING I/OS IN VIRTUALIZATION VIA SPDK VHOST SOLUTION

Changpeng Liu

Senior Storage Software Engineer

Intel Data Center Group

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/benchmarks .

Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

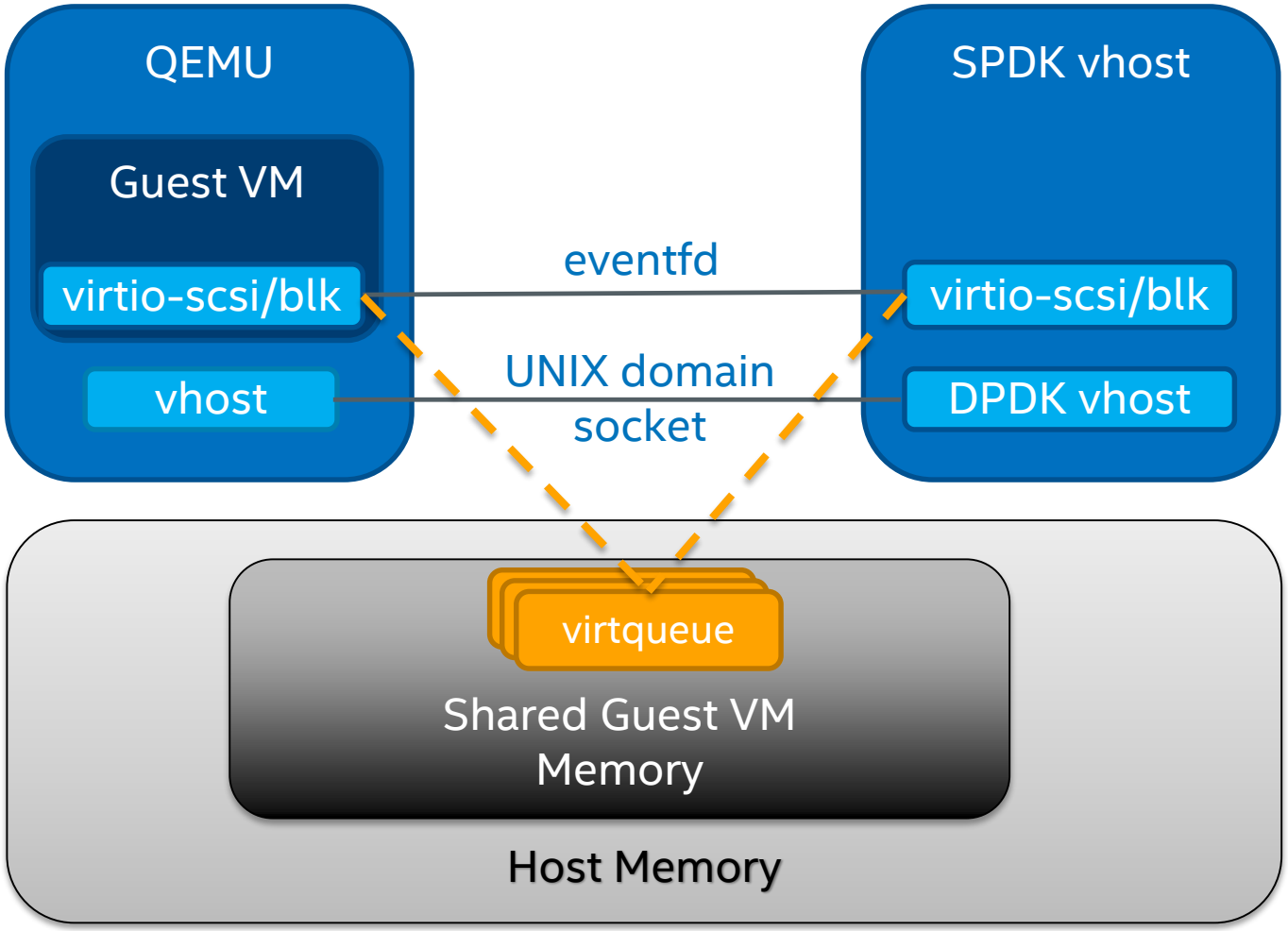
Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

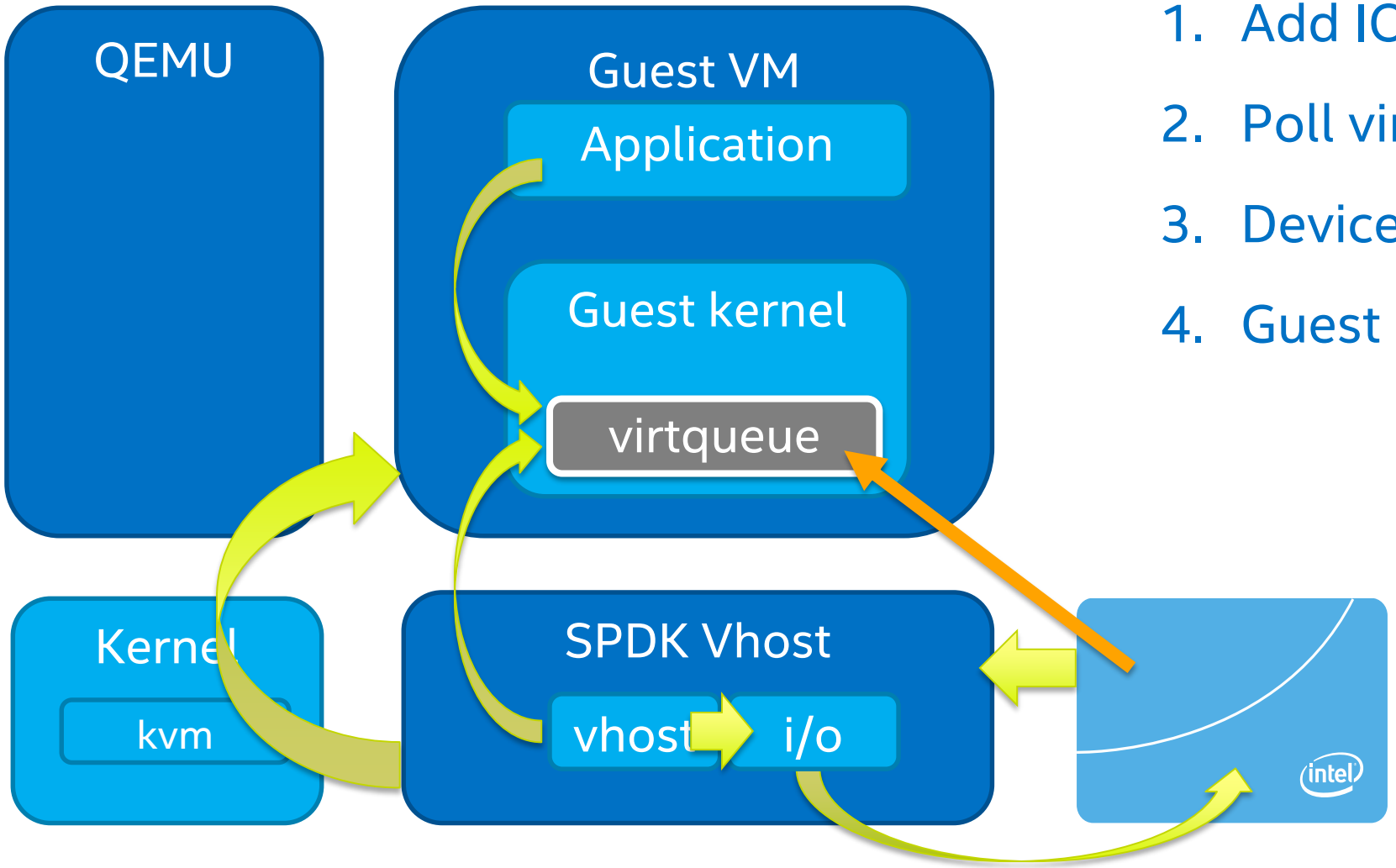
Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

INTRODUCTION

SPDK VHOST ARCHITECTURE

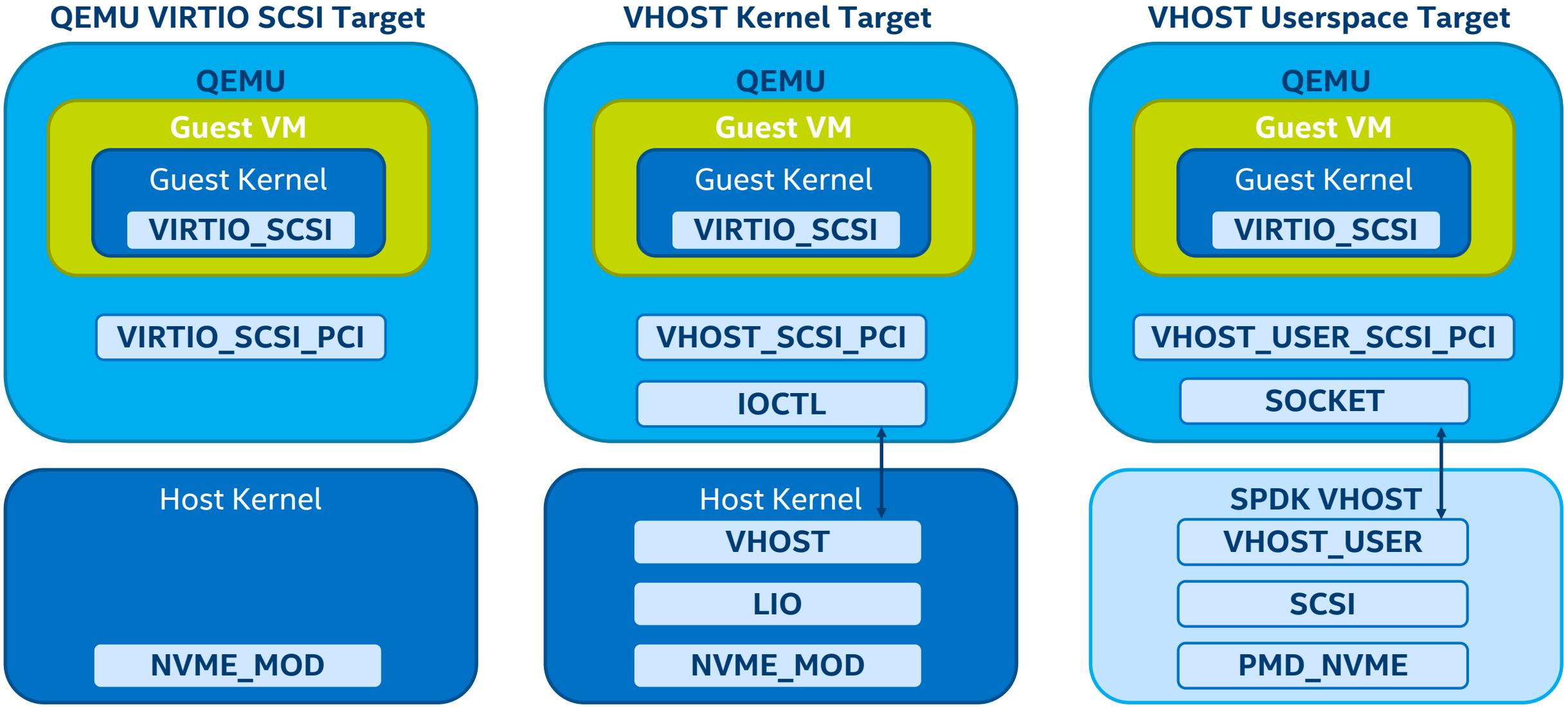


SPDK VHOST



1. Add IO to virtqueue
2. Poll virtqueue
3. Device executes IO
4. Guest completion interrupt

COMPARISON WITH EXISTING SOLUTIONS



SPDK VHOST Target Summary

Vhost Target	QEMU Support	Guest Support	Container similar Solution Support
Vhost SCSI Target	Yes	Yes, Kernel+PMD	Yes
Vhost Blk Target	Yes	Yes, Kernel+PMD	Yes
Vhost NVMe Target	No, SPDK QEMU branch	Yes, Kernel+PMD	No

-Vhost-SCSI: QEMU 2.9 added vhost-user-scsi-pci host driver support

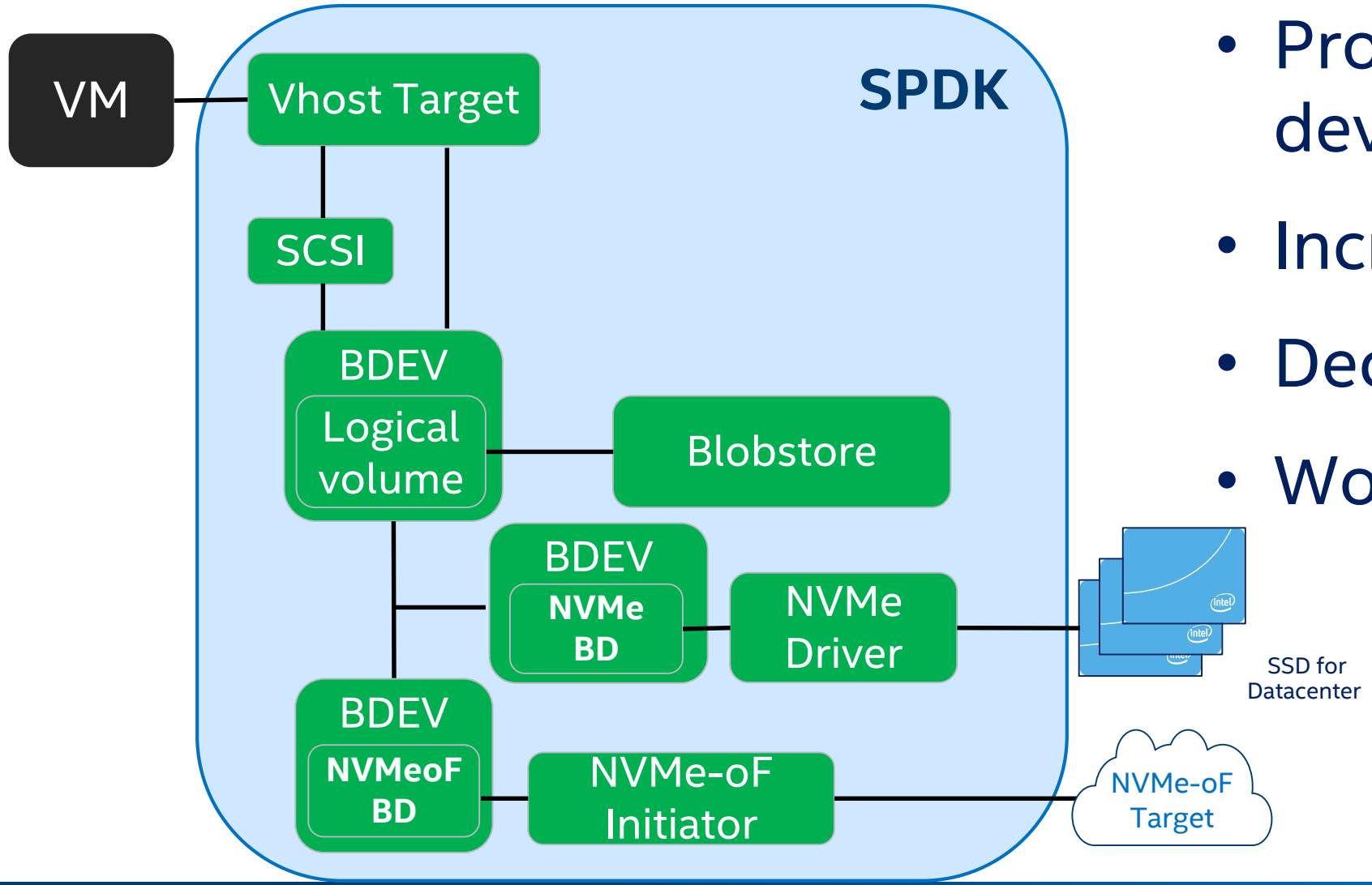
-Vhost-Blk: QEMU 2.11 added vhost-user-blk-pci host driver support

-Vhost NVMe: a new device type which can demonstrate NVMe controller to VM, native kernel NVMe driver can be used

USE CASES

Virtual Machine Acceleration

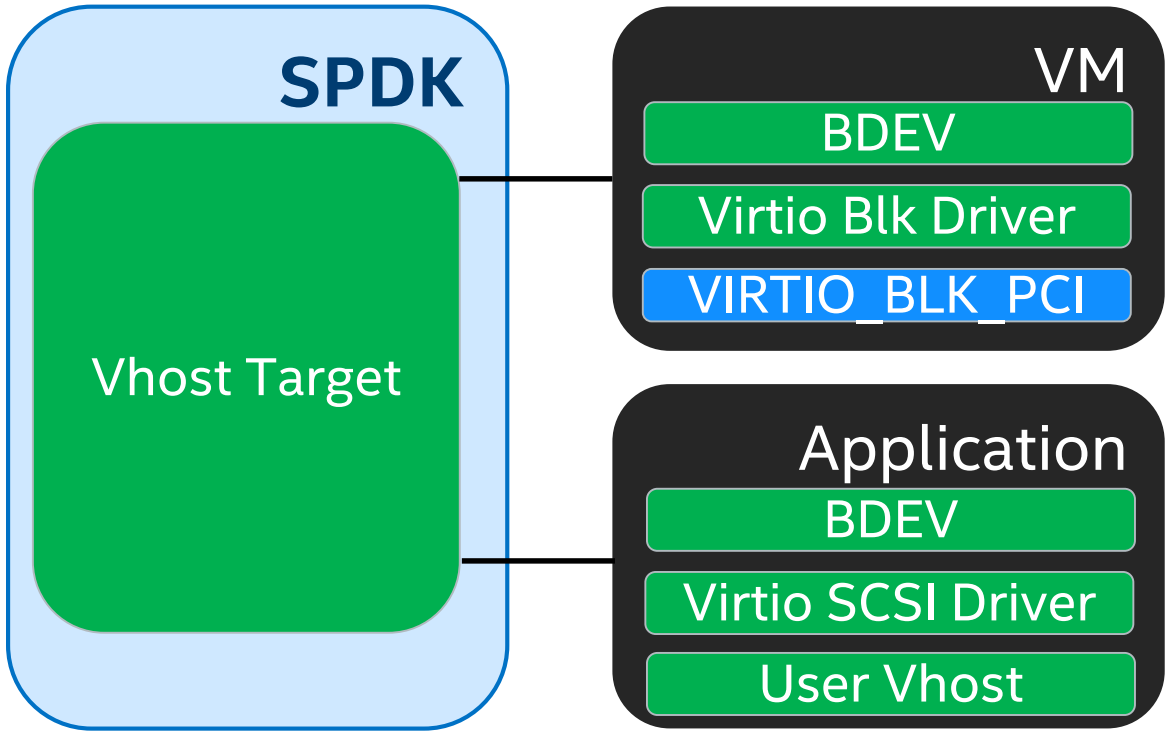
Released



- Provides dynamic block device provisioning
- Increase VM Density
- Decrease Guest Latency
- Works with KVM/QEMU

Virtio SCSI/blk Driver

Released



Virtio SCSI/Blk is an initiator for SPDK Vhost target

Virtio SCSI/Blk driver supports 2 usage models:

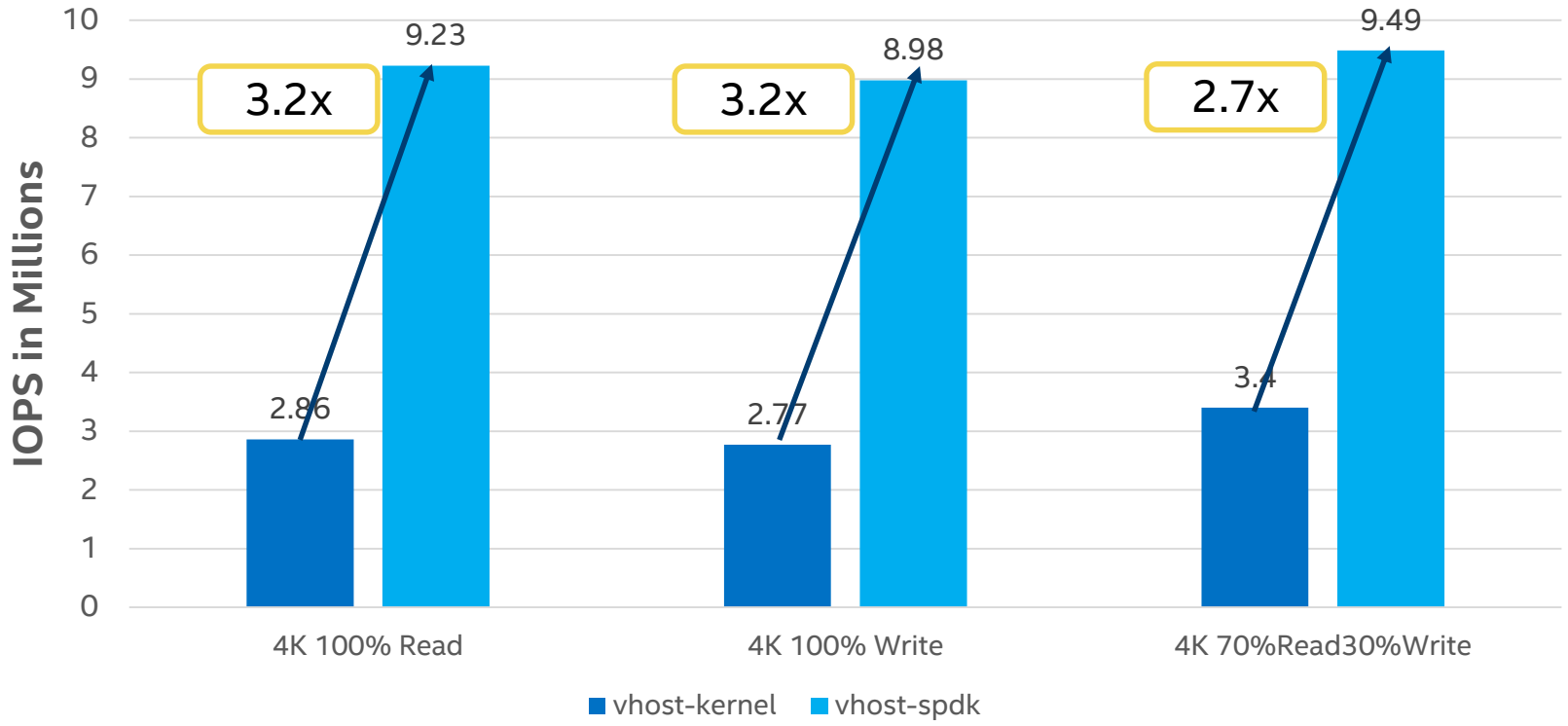
- PCI Mode: Polling mode driver inside Guest VM
- User vhost: Can be used to connect to vhost target directly via socket, e.g.: containers or multi-process application

BENCHMARKS

48 VMs: vhost-scsi performance (SPDK vs. Kernel)

Intel Xeon Platinum 8180 Processor, 24x Intel P4800x 375GB

2 partitions per VM, 10 vhost I/O processing cores

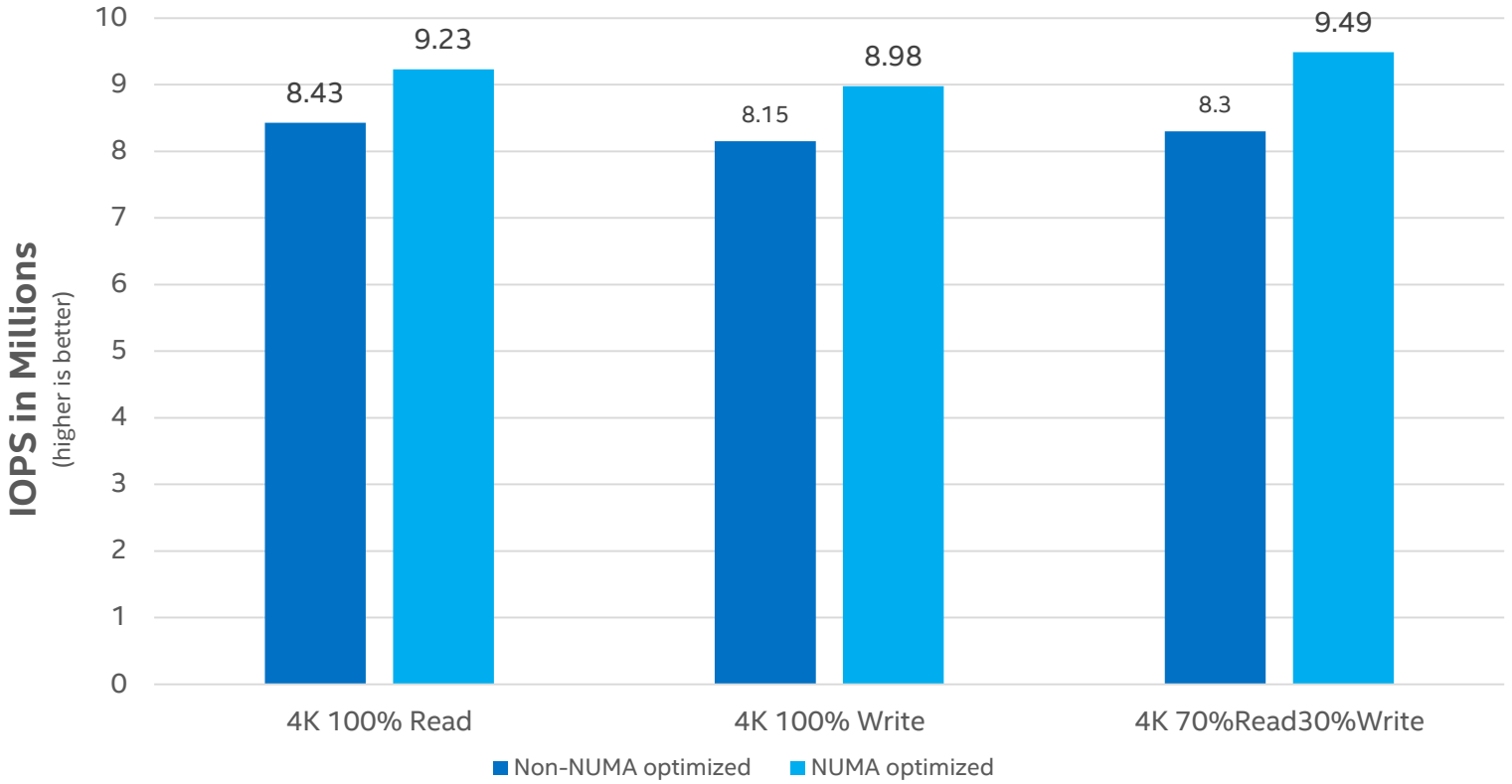


- Aggregate IOPS across all 48x VMs reported. All VMs on separate cores than vhost-scsi cores.
- 10 vhost-scsi cores for I/O processing
- SPDK vhost-scsi up to 3.2x better with 4K 100% Random read I/Os
- Used cgroups to restrict kernel vhost-scsi processes to 10 cores

System Configuration:Intel Xeon Platinum 8180 @ 2.5GHz. 56 physical cores 6x 16GB, 2667 DDR4, 6 memory Channels, SSD: Intel P4800x 375GB x24 drives, Bios: HT disabled, p-states enabled, turbo enabled, Ubuntu 16.04.1 LTS, 4.11.0 x86_64 kernel, 48 VMs, number of partition: 2, VM config : 1core 1GB memory, VM OS: fedora 25, blk-mq enabled, Software packages: Qemu-2.9, libvirt-3.0.0, spdk (3bfecec994), IO distribution: 10 vhost-cores for SPDK / Kernel. Rest 46 cores for QEMU using cgroups, FIO-2.1.10 with SPDK plugin, io depth=1, 8, 32 numjobs=1, direct=1, block size 4k

NUMA vs. Non-NUMA: SPDK vhost-scsi

Intel Xeon Platinum 8180 Processor, 24x Intel P4800x 375GB 48VMs, 10 vhost-scsi cores



- 10% performance improvement with NUMA optimized.
- NUMA optimization done to ensure vhost-scsi core match to NVMe drive socket location

System Configuration:Intel Xeon Platinum 8180 @ 2.5GHz. 56 physical cores 6x 16GB, 2667 DDR4, 6 memory Channels, SSD: Intel P4800x 375GB x24 drives, Bios: HT disabled, p-states enabled, turbo enabled, Ubuntu 16.04.1 LTS, 4.11.0 x86_64 kernel, 48 VMs, number of partition: 2, VM config : 1core 1GB memory, VM OS: fedora 25, blk-mq enabled, Software packages: Qemu-2.9, libvirt-3.0.0, spdk (3bfecec994), IO distribution: 10 vhost-cores for SPDK / Kernel. Rest 46 cores for QEMU using cgroups, FIO-2.1.10 with SPDK plugin, io depth=1, 8, 32 numjobs=1, direct=1, block size 4k

UPDATE

Virtio-blk protocol extension

- **Faster than virtio-scsi protocol due to eliminate SCSI middle layer inside Guest kernel**
- **Linux Block layer supports multi-queues for virtio-blk**
- **Lack of support for DISCARD/WRITE ZEROES commands.**
- **Virtio-Blk protocol specification has added this feature.**

See https://github.com/oasis-tcs/virtio-spec for reference. Linux kernel driver and QEMU driver will be kicked soon.

Vhost-NVMe

- **What's vhost-nvme?**

NVMe specification as the communication protocol between Guest and slave I/O target

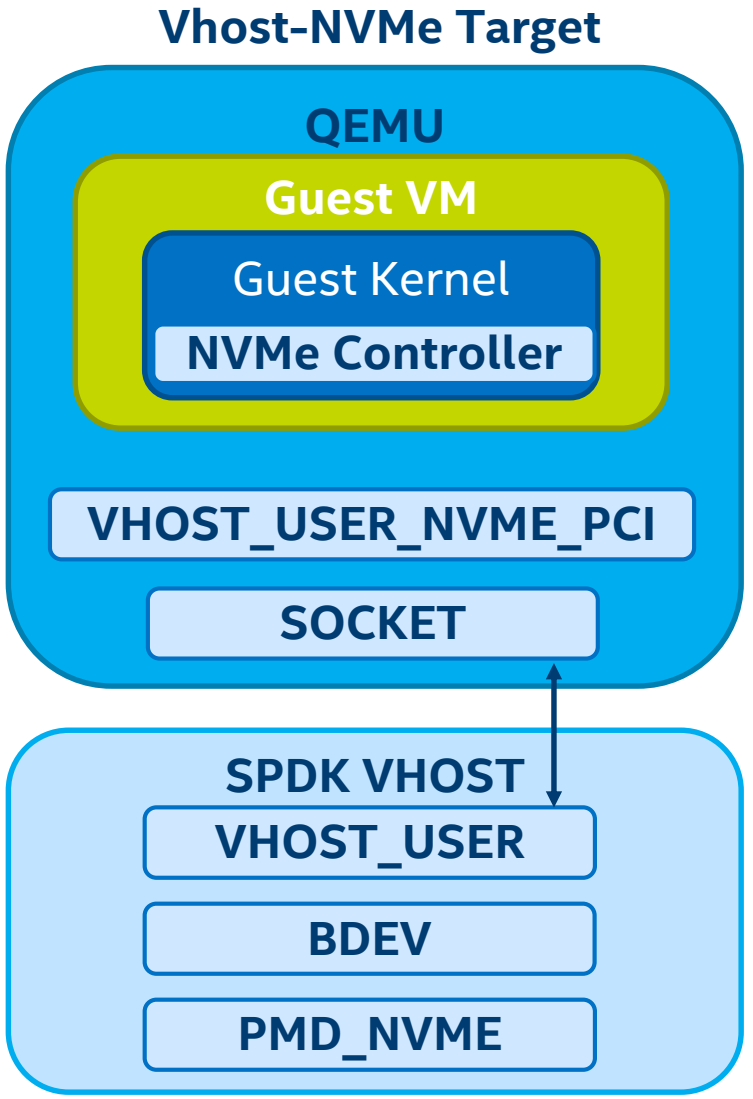
Make use of UNIX domain socket as the message channel to setup I/O queues and interrupt notifier for Guest

NVMe 1.3 specification virtualization enhancement

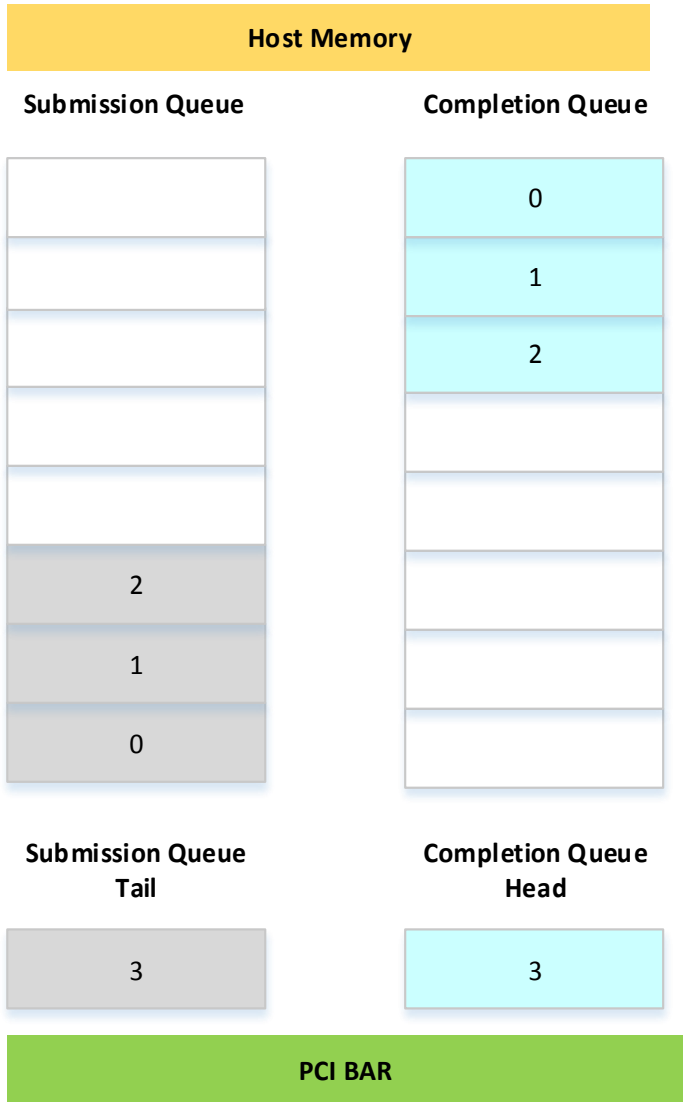
- **What's the benefit?**

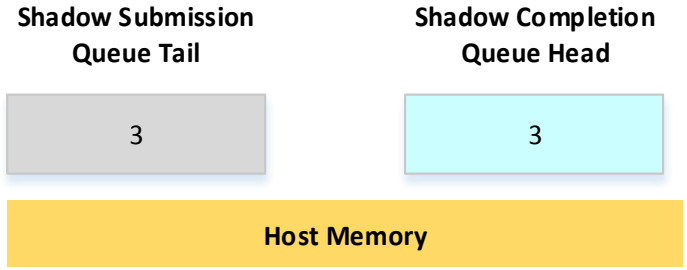
Native kernel NVMe driver can be used inside VM without any extra modifications

Eliminate the SCSI middle layer driver compared with exist vhost scsi solution, which can improve the performance



NVMe 1.3 Specification Enhancement





NVMe 1.3 Virtualization Enhancement:

1. Optional Admin Command Support
2. Doorbell Buffer Config

Vhost-NVMe Implementation

Vhost Message Protocol	Description
Get Controller Capabilities	Controller capabilities register of NVMe specification
Get Device ID	Vendor ID of the emulated NVMe controller of QEMU
Get/Set Controller Configuration	Enable/Disable emulated NVMe controller
Admin Command Pass-through	Admin commands routed to slave target
Set Memory Table	Sets the memory map regions on the slave target so it can translate the I/O queues' addresses.
Set Guest Notifier	Set the event file descriptor for the purpose to interrupt the Guest when I/O is completed.
Set Event Notifier	Set the event file descriptor for AER.

Table 1: Vhost socket messages

Admin Commands	Description
Identify/Identify NS	QEMU gets the identify data from slave target, QEMU can cache it with a local copy to avoid repeated vhost messages.
Create/Delete Submission Queue	QEMU allocates/deletes the queues and send the Admin command to slave target.
Create/Delete Completion Queue	Each Create Completion Queue command should follow with a Set Guest Notifier for IRQ notification.
Abort	Slave target will process Abort command.
Asynchronous Event Request	QEMU sends the command to slave target and follows with a Set Event Notifer for real AER.
Doorbell Buffer Config	Set the shadow doorbell buffer in slave target

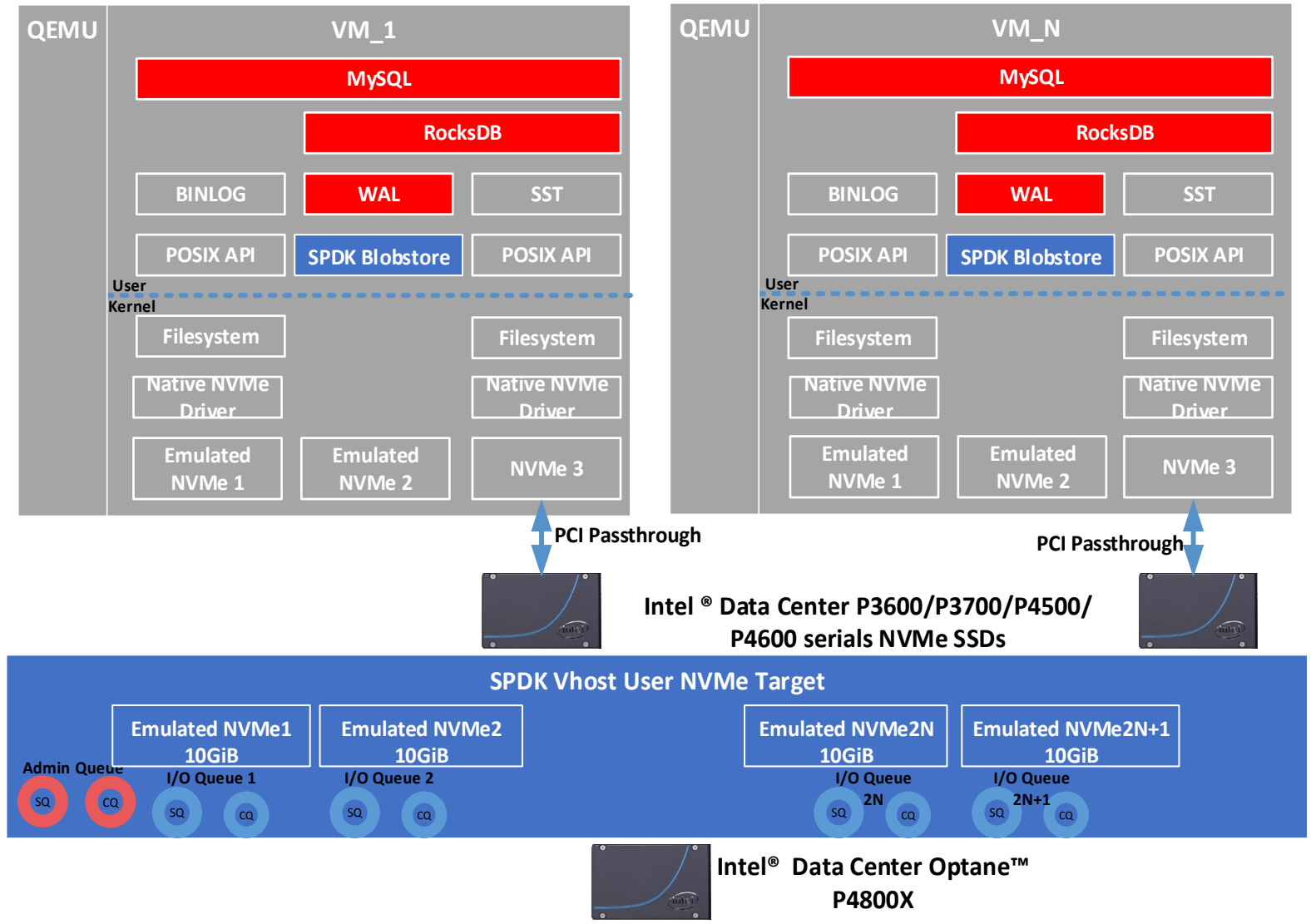
Table 2: Mandatory Admin commands in slave target

Use Cases

- **Integrating SPDK Blobstore with RocksDB to MySQL inside VM**

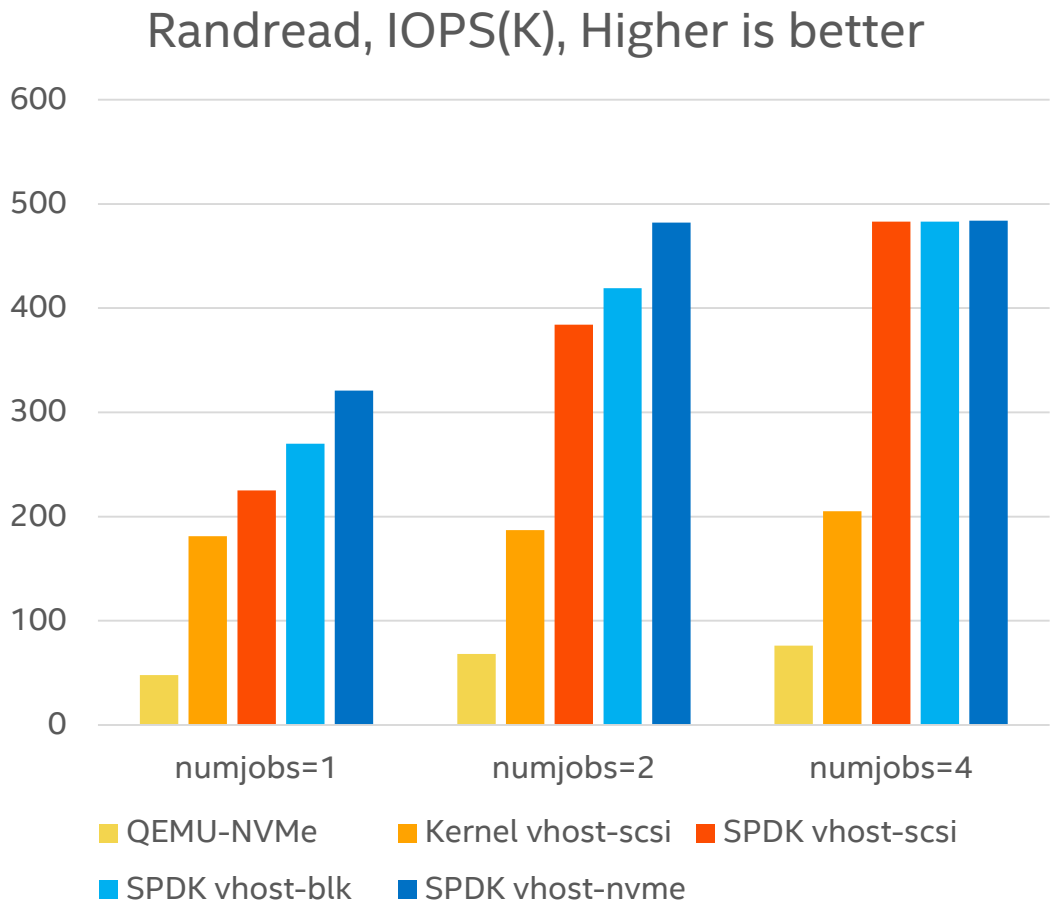
-Optane™ can be parted into several logical volumes to each VM for critical log usage.

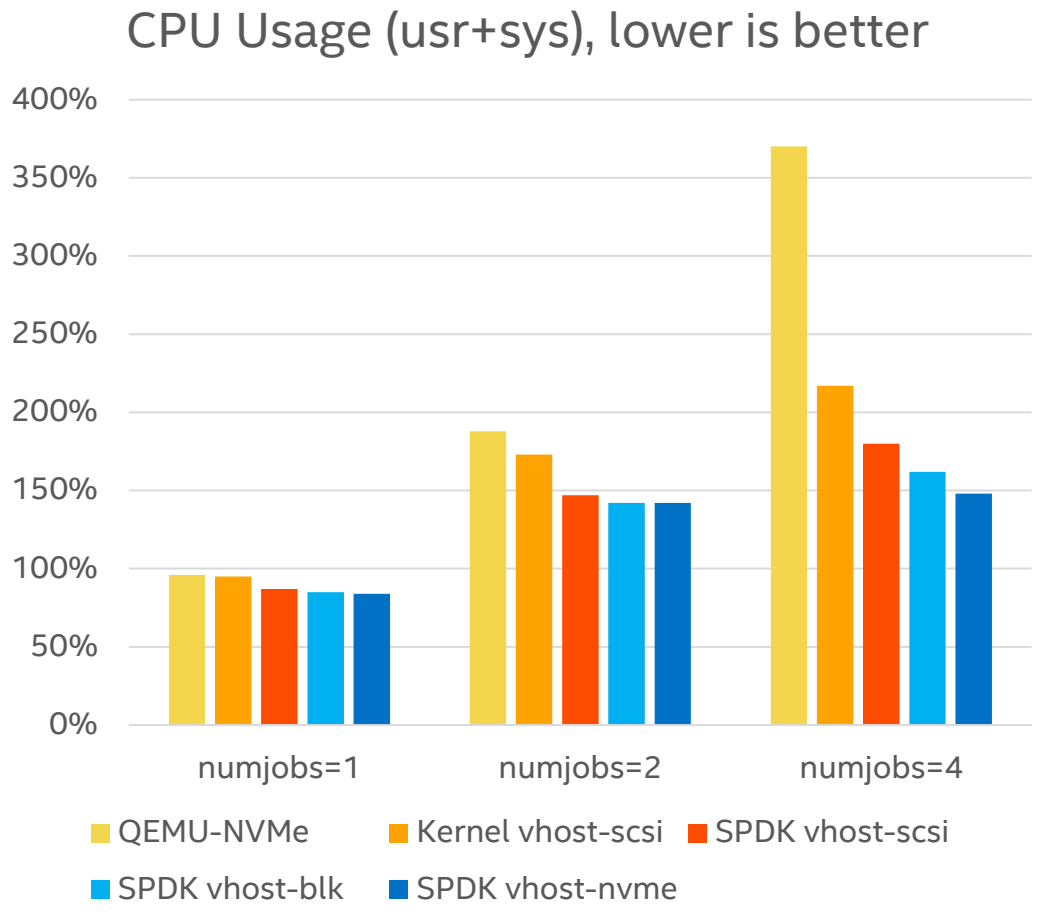
-Enable WAL with SPDK to provide short I/O path without any data copies.



Benchmarks

1 VM with 1 NVMe SSD, 4 VCPU

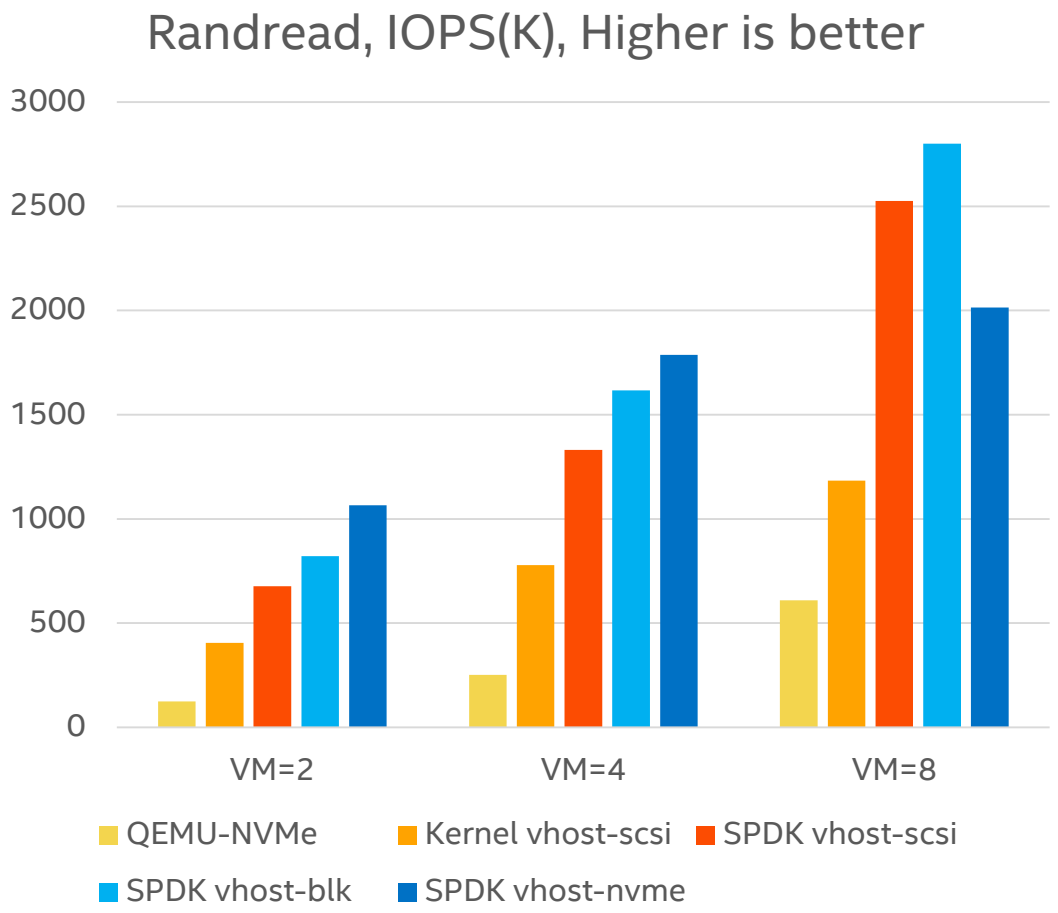




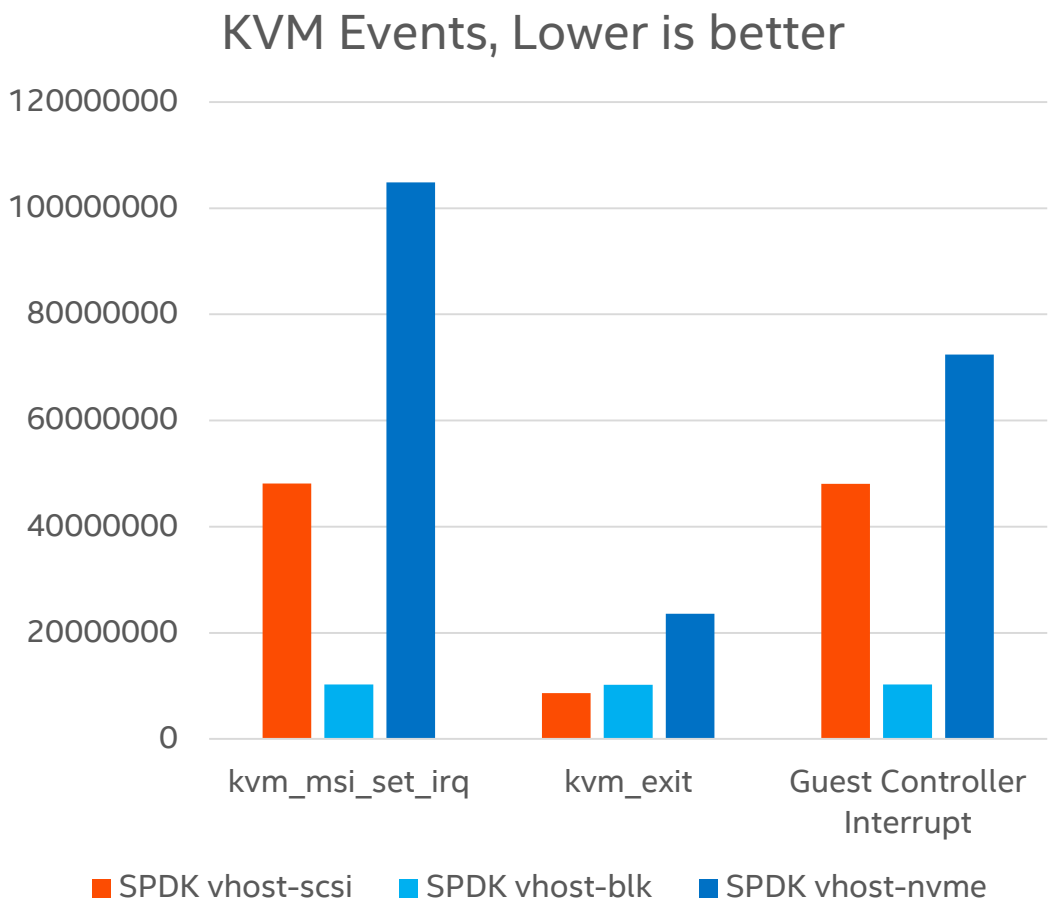
System Configuration: 2 * Intel Xeon E5 2699v4 @ 2.2GHz, 128GB, 2667 DDR4, 6 memory Channels, SSD: Intel P3700 800GB, FW: 8DV101H0, Bios: HT disabled, CentOS 7.4(kernel 4.12.5), 1 VMs, VM config : 4core 4GB memory, VM OS: Fedora 25(kernel 4.14.0), blk-mq enabled, Software packages: Qemu-2.11,IO distribution: 1 vhost-cores for SPDK, FIO, io depth=128 numjobs=1,2,4 direct=1, block size 4k.

Benchmarks and KVM Events

8 VMs shared 4 NVMe SSD, 4 VCPU



1 VMs with 1 NVMe SSD, 4 VCPU



System Configuration: 2 * Intel Xeon E5 2699v4 @ 2.2GHz, 128GB, 2667 DDR4, 6 memory Channels, SSD: Intel P4510 2TB, FW: QDV1013A, Bios: HT disabled, CentOS 7.4(kernel 4.12.5), 1 VMs, VM config : 4core 4GB memory, VM OS: Fedora 25(kernel 4.14.0), blk-mq enabled, Software packages: Qemu-2.11,IO distribution: 1 vhost-cores for SPDK, FIO, io depth=128**,** numjobs=2; FIO, io depth=64 numjobs=4, size=100GB; direct=1 block size 4k.
block size 4k.


intel
experience
what's inside™