

SPDK Vhost Performance Report

Release 24.05

Testing Date: July 2024

Performed by:

Karol Latecki (karol.latecki@intel.com)

Jaroslav Chachulski (jaroslawx.chachulski@intel.com)

Acknowledgments:

Krzysztof Karaś (krzysztof.karas@intel.com)

Michał Berger (michal.berger@intel.com)

Contents

| | |
|---|----|
| Contents | 2 |
| Audience and Purpose..... | 3 |
| Test setup | 4 |
| Hardware configuration | 4 |
| BIOS Settings | 5 |
| Virtual Machine Settings..... | 5 |
| Introduction to the SPDK Vhost target | 7 |
| SPDK Vhost target architecture | 7 |
| Test Case 1: SPDK Vhost Core Scaling | 9 |
| 4KiB Random Read Results..... | 11 |
| 4KiB Random Write Results | 12 |
| 4KiB Random Read-Write Results | 13 |
| Logical Volumes performance impact | 14 |
| Packed Ring performance impact..... | 15 |
| Conclusions | 16 |
| Test Case 2: Rate Limiting IOPS per VM..... | 17 |
| Test Case 2 Results | 19 |
| Conclusions | 21 |
| Test Case 3: Performance per NVMe drive | 21 |
| Test Case 3 results..... | 23 |
| Conclusions | 26 |
| Summary | 27 |
| List of Tables | 28 |
| List of Figures | 29 |

Audience and Purpose


This report is intended for people who are interested in looking at SPDK Vhost-Scsi and Blk stack performance and comparison to its Linux kernel equivalents. It provides performance and efficiency comparisons between SPDK Vhost-Scsi and Linux Kernel Vhost-Scsi software stacks under various test cases.

The purpose of this report is not to imply a single correct approach, but rather to provide a baseline of well-tested configurations and procedures that produce repeatable and reproducible results. This report can also be viewed as information regarding best known method when performance testing SPDK Vhost-Scsi and Vhost-Blk stacks.

Test setup

Hardware configuration

Table 1: Hardware setup configuration

| Item | Description | | | | | | | | | | | | | | | | | | |
|----------------------|--|----|----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Server Platform | Ultra SuperServer SYS-220U-TNR  | | | | | | | | | | | | | | | | | | |
| Motherboard | Server board X12DPU-6 | | | | | | | | | | | | | | | | | | |
| CPU | 2 CPU sockets, Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz Number of cores 28 per socket, number of threads 56 per socket Both sockets populated. Microcode: 0xd0003d1 | | | | | | | | | | | | | | | | | | |
| Memory | 16 x 32GB SK Hynix DDR4 HMA84GR7DJR4N-XN; Total 512 GBs. Memory channel population: <table border="1" data-bbox="495 1119 1446 1493"> <thead> <tr> <th>P1</th> <th>P2</th> </tr> </thead> <tbody> <tr> <td>CPU1_DIMM_A1</td> <td>CPU2_DIMM_A1</td> </tr> <tr> <td>CPU1_DIMM_B1</td> <td>CPU2_DIMM_B1</td> </tr> <tr> <td>CPU1_DIMM_C1</td> <td>CPU2_DIMM_C1</td> </tr> <tr> <td>CPU1_DIMM_D1</td> <td>CPU2_DIMM_D1</td> </tr> <tr> <td>CPU1_DIMM_E1</td> <td>CPU2_DIMM_E1</td> </tr> <tr> <td>CPU1_DIMM_F1</td> <td>CPU2_DIMM_F1</td> </tr> <tr> <td>CPU1_DIMM_G1</td> <td>CPU2_DIMM_G1</td> </tr> <tr> <td>CPU1_DIMM_H1</td> <td>CPU2_DIMM_H1</td> </tr> </tbody> </table> | P1 | P2 | CPU1_DIMM_A1 | CPU2_DIMM_A1 | CPU1_DIMM_B1 | CPU2_DIMM_B1 | CPU1_DIMM_C1 | CPU2_DIMM_C1 | CPU1_DIMM_D1 | CPU2_DIMM_D1 | CPU1_DIMM_E1 | CPU2_DIMM_E1 | CPU1_DIMM_F1 | CPU2_DIMM_F1 | CPU1_DIMM_G1 | CPU2_DIMM_G1 | CPU1_DIMM_H1 | CPU2_DIMM_H1 |
| P1 | P2 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_A1 | CPU2_DIMM_A1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_B1 | CPU2_DIMM_B1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_C1 | CPU2_DIMM_C1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_D1 | CPU2_DIMM_D1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_E1 | CPU2_DIMM_E1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_F1 | CPU2_DIMM_F1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_G1 | CPU2_DIMM_G1 | | | | | | | | | | | | | | | | | | |
| CPU1_DIMM_H1 | CPU2_DIMM_H1 | | | | | | | | | | | | | | | | | | |
| Operating System | Fedora 37 | | | | | | | | | | | | | | | | | | |
| BIOS | 1.8 | | | | | | | | | | | | | | | | | | |
| Linux kernel version | 6.1.6-200.fc37.x86_64 Spectre-meltdown mitigations enabled | | | | | | | | | | | | | | | | | | |
| SPDK version | SPDK 24.05 | | | | | | | | | | | | | | | | | | |
| Qemu version | 7.0.0 (qemu-7.0.0-12.fc37) | | | | | | | | | | | | | | | | | | |
| Storage | OS: 1x 250GB Crucial CT250MX500SSD1 Storage: 22x Kioxia® KCM61VUL3T20 3.2TBs (FW: 0105) (10 on CPU NUMA Node 0, 12 on CPU NUMA Node 1) | | | | | | | | | | | | | | | | | | |

BIOS Settings

Table 2: Test platform BIOS settings

| Item | Description |
|------|--|
| BIOS | VT-d = Enabled CPU Power and Performance Policy = <Performance> CPU C-state = No Limit CPU P-state = Enabled Enhanced Intel® Speedstep® Tech = Enabled Turbo Boost = Enabled Hyper Threading = Enabled |

Table 3: Test System NVMe storage setup

| Item | Description | |
|--|---|-------------------------------------|
| PCIe Riser cards | <p>“Ultra” Riser Card: AOC-2UR68G4-i2XT</p> <ul style="list-style-type: none"> • PCIe Slot 1 – x16, CPU2 • PCIe Slot 2 – x8, CPU2 • PCIe Slot 3 – x8, CPU2 <p>Right-facing riser card: RSC-WR-6</p> <ul style="list-style-type: none"> • PCIe Slot 4 – x16, CPU1 <p>Left-facing riser card: RSC-W2-66G4</p> <ul style="list-style-type: none"> • PCIe Slot 5 – x16, CPU2 • PCIe Slot 7 – x16, CPU1 <p>More information can be found in SYS-220U-TNR manual document.</p> | |
| PCIe Retimer cards | 3 x AOC-SLG4-4E4T Installed in: <ul style="list-style-type: none"> ○ PCIe Retimer 1: RSC-WR-6, PCIe Slot 4 (using CPU1 PCIe Lanes) ○ PCIe Retimer 2: AOC-2UR68G4-i2XT, PCIe Slot 1 (using CPU2 PCIe Lanes) ○ PCIe Retimer 3: RSC-W2-66G4, PCIe Slot 5 (using CPU2 PCIe Lanes) | |
| NVMe Drives distribution across the system | Nvme0 – 5 | Motherboard ports (CPU1 PCIe Lanes) |
| | Nvme6 – 9 | Motherboard ports (CPU2 PCIe Lanes) |
| | Nvme9 – 13 | PCIe Retimer 1 (CPU1 PCIe Lanes) |
| | Nvme14 - 17 | PCIe Retimer 2 (CPU2 PCIe Lanes) |
| | Nvme18 - 21 | PCIe Retimer 3 (CPU2 PCIe Lanes) |

Virtual Machine Settings

Table 4: Guest VM configuration

| Item | Description |
|------|---|
| CPU | 2 vCPU, pass through from physical host server. Explicit core usage enforced using “taskset –a –c” command. QEMU arguments for starting the VM: -cpu host -smp 1 |

| | |
|---|---|
| Memory | <p>2 GB RAM. Memory is pre-allocated for each VM using Hugepages on host system and used from appropriate NUMA node, to match the CPU which was passed to the VM.</p> <p>QEMU arguments: -m 2048 -object memory-backend-file,id=mem,size=2048M,mem-path=/dev/hugepages,share=on,prealloc=yes,host-nodes=0,policy=bind</p> |
| Operating System | <p>Fedora 35</p> |
| Linux kernel version | <p>5.15.7-200.fc35.x86_64</p> |
| Additional boot options in /etc/default/grub | <p>Multi queue enabled: scsi_mod.use_blk_mq=1</p> |

Introduction to the SPDK Vhost target

SPDK Vhost is a userspace target designed to extend the performance efficiencies of SPDK into QEMU/KVM virtualization environments. The SPDK Vhost-Scsi target presents a broad range of SPDK-managed block devices into virtual machines. SPDK community has leveraged existing SPDK SCSI layer, DPDK Vhost library, QEMU Vhost-Scsi and Vhost-Blk functionality to create the high performance SPDK userspace Vhost target.

SPDK Vhost target architecture

QEMU sets up the Vhost target via UNIX domain socket. QEMU pre-allocates huge pages for the guest VM to enable DMA by the Vhost target. The guest VM submits I/O directly to the Vhost target via virtqueues in shared memory as shown in Figure 1. The Vhost target transfers data to/from the guest VM via shared memory. The Vhost target then completes I/O to the guest VM via virtqueues in shared memory. There is a completion interrupt sent using eventfd which requires a system call and a guest VM exit. It should be noted that there is no QEMU intervention during the I/O submission process.

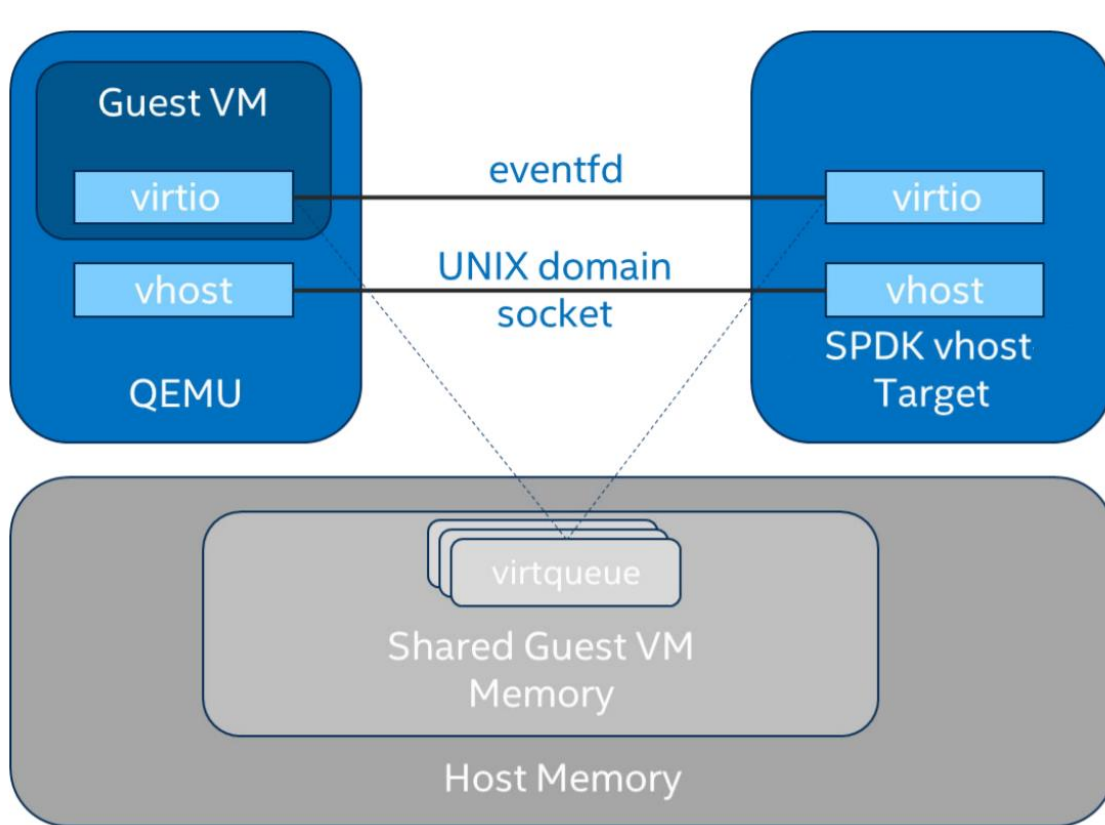


Figure 1: SPDK Vhost-Scsi architecture

This report shows the performance comparisons between the traditional interrupt-driven Linux Kernel Vhost-Scsi and the accelerated polled-mode SPDK Vhost-Scsi under 3 different test cases using local NVMe storage. Additionally, the SPDK Vhost-Blk stack is included in the report for further comparison with the SCSI stack.

Test Case 1: SPDK Vhost Core Scaling

This test case was performed to understand aggregate VM performance with SPDK Vhost I/O core scaling. We ran up to 36 virtual machines, each running following FIO workloads:

- 4KiB 100% Random Read
- 4KiB 100% Random Write
- 4KiB Random 70% Read / 30 % Write

We increased the number of CPU cores used by SPDK Vhost target to process I/O from 1 up to 22 and measured the throughput (in IOPS) and latency. The number of VMs between test runs was not constant and was increased by 2 for each Vhost CPU added, up to a maximum of 44 VMs. VM number was not increased beyond 44, because maximum available CPU core number was reached.

FIO was run in client-server mode. FIO client was run on the host machine and distributed jobs to FIO servers run on each VM. This allowed us to start the FIO jobs across all VMs at the same time.

Results in the table and charts represent aggregate performance (IOPS and average latency) seen across all the VMs. The results are average of 3 runs.

Table 5: SPDK Vhost Core Scaling test configuration

| Item | Description |
|---------------------------|---|
| Test case | Test SPDK Vhost target I/O core scaling performance |
| Test configuration | <p>FIO Version: fio-3.28</p> <p>VM Configuration:</p> <ul style="list-style-type: none"> • Common settings are described in the Virtual Machine Settings chapter. • Number of VMs: variable (2 VMs per 1 Vhost CPU core, up to 44 VMs max). • Each VM has a single Vhost device as a target for the FIO workload. This is achieved by sharing SPDK NVMe bdevs by using either a Split NVMe vbdev or Logical Volume bdev configuration. <p>SPDK Vhost target configuration:</p> <ul style="list-style-type: none"> • Test were run with both the Vhost-Scsi and Vhost-Blk stacks. • The Vhost-Scsi stack was run with Split NVMe bdevs and Logical Volume bdevs. • Vhost-Blk stack was run with Logical Volume bdevs. • Tests were performed with 1, 2, 6, 10, 14, 18 and 22 Vhost cores for each stack-bdev combination. <p>Kernel Vhost target configuration:</p> <ul style="list-style-type: none"> • N/A |
| FIO configuration | [global] ioengine=libaio direct=1 |

```
thread=1
norandommap=1
time_based=1
gtod_reduce=0
ramp_time=60s
runtime=240s
numjobs=2
bs=4k
rw=randrw
rwmixread=100 (100% reads), 70 (70% reads, 30% writes), 0 (100% writes)
iodepth= {1, 64, 128, 192, 384}
```

4KiB Random Read Results

Table 6: SPDK Vhost core scaling results, 4KiB 100% Random Reads IOPS, QD=64

| # of CPU cores | # of VMs | Stack / Backend | IOPS (millions) |
|----------------|----------|------------------------|-----------------|
| 1 | 2 | SCSI / Split NVMe Bdev | 1.71 |
| | | SCSI / Lvol Bdev | 1.71 |
| | | BLK / Lvol Bdev | 1.79 |
| 2 | 4 | SCSI / Split NVMe Bdev | 3.39 |
| | | SCSI / Lvol Bdev | 3.39 |
| | | BLK / Lvol Bdev | 3.58 |
| 6 | 12 | SCSI / Split NVMe Bdev | 9.66 |
| | | SCSI / Lvol Bdev | 9.39 |
| | | BLK / Lvol Bdev | 8.14 |
| 10 | 20 | SCSI / Split NVMe Bdev | 15.29 |
| | | SCSI / Lvol Bdev | 14.05 |
| | | BLK / Lvol Bdev | 13.10 |
| 14 | 28 | SCSI / Split NVMe Bdev | 15.78 |
| | | SCSI / Lvol Bdev | 14.27 |
| | | BLK / Lvol Bdev | 13.09 |
| 18 | 36 | SCSI / Split NVMe Bdev | 17.80 |
| | | SCSI / Lvol Bdev | 17.06 |
| | | BLK / Lvol Bdev | 14.35 |
| 22 | 44 | SCSI / Split NVMe Bdev | 19.32 |
| | | SCSI / Lvol Bdev | 19.36 |
| | | BLK / Lvol Bdev | 16.00 |

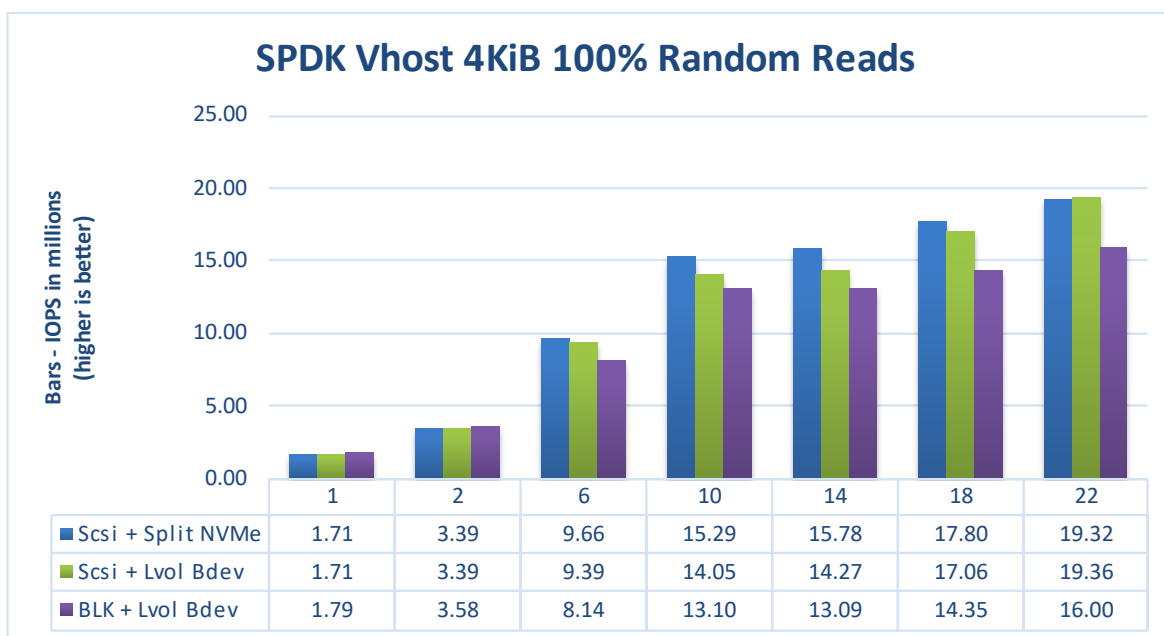


Figure 2: Comparison of performance between various SPDK Vhost stack-bdev combinations for 4KiB Random Read QD=64 workload

4KiB Random Write Results

Table 7: SPDK Vhost core scaling results, 4KiB 100% Random Write IOPS, QD=64

| # of CPU cores | # of VMs | Stack / Backend | IOPS (millions) |
|----------------|----------|------------------------|-----------------|
| 1 | 2 | SCSI / Split NVMe Bdev | 1.08 |
| | | SCSI / Lvol Bdev | 1.08 |
| | | BLK / Lvol Bdev | 1.12 |
| 2 | 4 | SCSI / Split NVMe Bdev | 2.48 |
| | | SCSI / Lvol Bdev | 2.48 |
| | | BLK / Lvol Bdev | 2.64 |
| 6 | 12 | SCSI / Split NVMe Bdev | 7.13 |
| | | SCSI / Lvol Bdev | 7.16 |
| | | BLK / Lvol Bdev | 7.14 |
| 10 | 20 | SCSI / Split NVMe Bdev | 12.30 |
| | | SCSI / Lvol Bdev | 12.68 |
| | | BLK / Lvol Bdev | 12.58 |
| 14 | 28 | SCSI / Split NVMe Bdev | 13.10 |
| | | SCSI / Lvol Bdev | 12.42 |
| | | BLK / Lvol Bdev | 12.16 |
| 18 | 36 | SCSI / Split NVMe Bdev | 12.37 |
| | | SCSI / Lvol Bdev | 12.32 |
| | | BLK / Lvol Bdev | 11.46 |
| 22 | 44 | SCSI / Split NVMe Bdev | 12.67 |
| | | SCSI / Lvol Bdev | 12.67 |
| | | BLK / Lvol Bdev | 10.74 |

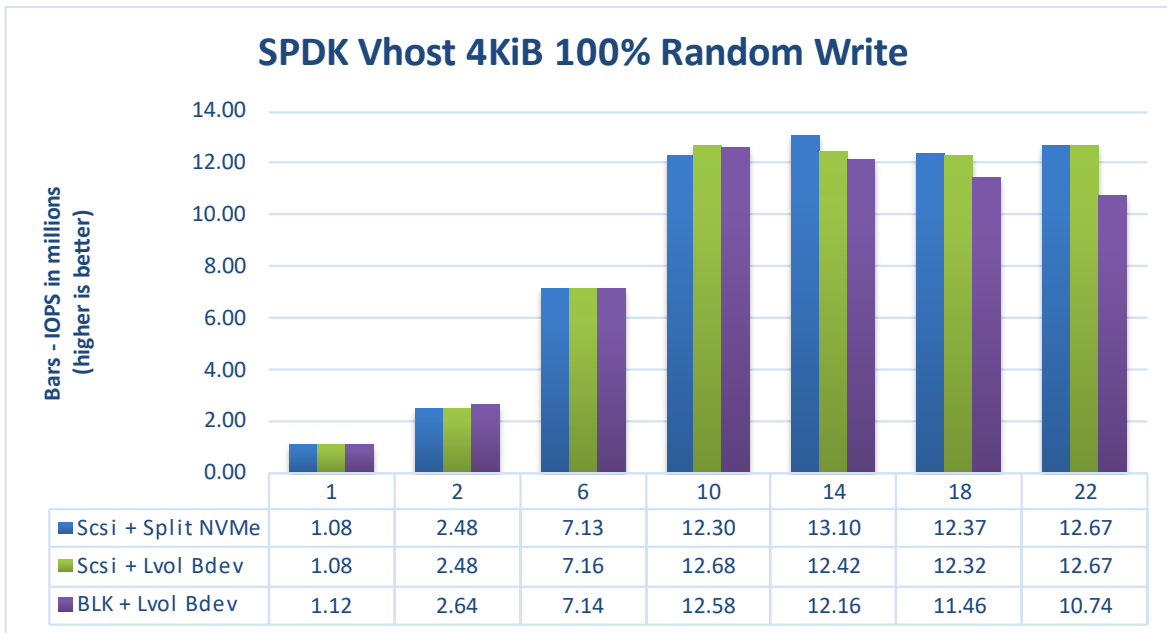


Figure 3: Comparison of performance between various SPDK Vhost stack-bdev combinations for 4KiB Random Write QD=64 workload

4KiB Random Read-Write Results

Table 8: SPDK Vhost core scaling results, 4KiB Random 70% Read 30% Write IOPS, QD=64

| # of CPU cores | # of VMs | Stack / Backend | IOPS (millions) |
|----------------|----------|------------------------|-----------------|
| 1 | 2 | SCSI / Split NVMe Bdev | 1.34 |
| | | SCSI / Lvol Bdev | 1.36 |
| | | BLK / Lvol Bdev | 1.46 |
| 2 | 4 | SCSI / Split NVMe Bdev | 2.81 |
| | | SCSI / Lvol Bdev | 2.90 |
| | | BLK / Lvol Bdev | 2.96 |
| 6 | 12 | SCSI / Split NVMe Bdev | 8.10 |
| | | SCSI / Lvol Bdev | 7.95 |
| | | BLK / Lvol Bdev | 7.44 |
| 10 | 20 | SCSI / Split NVMe Bdev | 13.15 |
| | | SCSI / Lvol Bdev | 12.56 |
| | | BLK / Lvol Bdev | 12.24 |
| 14 | 28 | SCSI / Split NVMe Bdev | 13.87 |
| | | SCSI / Lvol Bdev | 12.81 |
| | | BLK / Lvol Bdev | 12.18 |
| 18 | 36 | SCSI / Split NVMe Bdev | 15.37 |
| | | SCSI / Lvol Bdev | 14.60 |
| | | BLK / Lvol Bdev | 13.27 |
| 22 | 44 | SCSI / Split NVMe Bdev | 16.33 |
| | | SCSI / Lvol Bdev | 16.17 |
| | | BLK / Lvol Bdev | 14.79 |

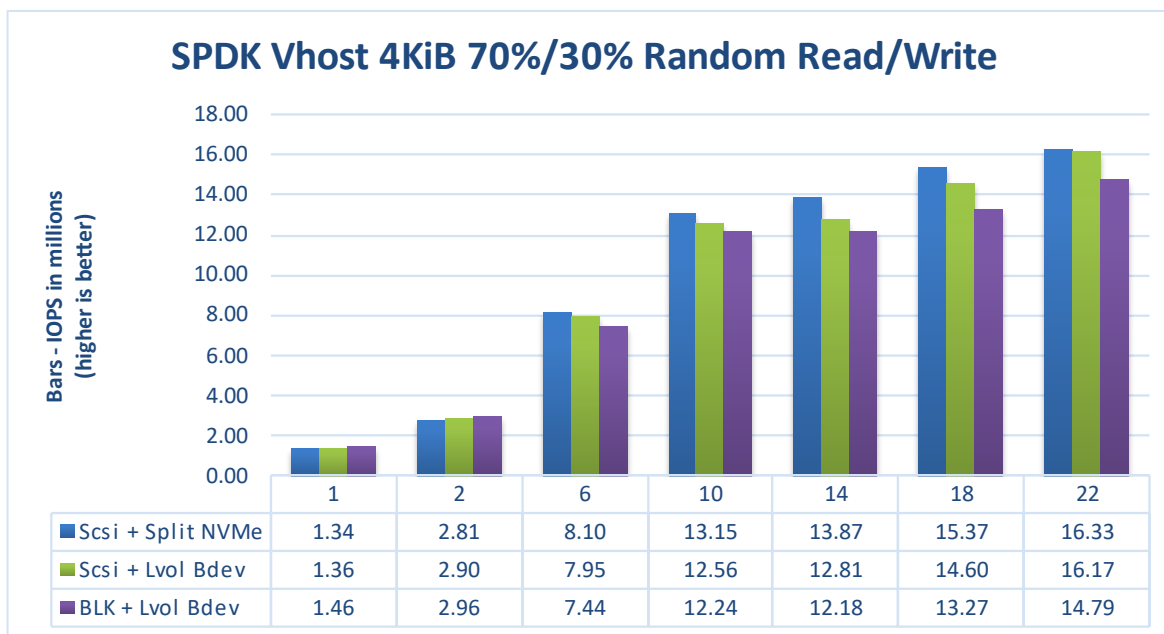


Figure 4: Comparison of performance between various SPDK Vhost stack-bdev combinations for 4KiB Random 70% Read 30% Write QD=64 workload

Logical Volumes performance impact

The SPDK Vhost-Scsi tests were run using two bdev backends – Split NVMe and Logical Volumes. Both “Split NVMe Bdevs” and “Logical Volume Bdevs” allow to logically partition NVMe SSDs, the latter being more flexible in configuration. Here we measure the overhead of extra flexibility afforded by Logical Volumes.

Table 9: Logical Volumes performance impact for SPDK Vhost SCSI

| Workload | # of CPU cores | # of VMs | Vhost SCSI + Split NVMe IOPS (millions) | Vhost SCSI + Lvol IOPS (millions) | Lvol Impact (%) |
|--|----------------|----------|---|-----------------------------------|-----------------|
| 4KiB 100% Random Read | 1 | 2 | 1.713 | 1.705 | -0.46% |
| | 2 | 4 | 3.395 | 3.393 | -0.05% |
| | 6 | 12 | 9.661 | 9.386 | -2.85% |
| | 10 | 20 | 15.286 | 14.046 | -8.11% |
| | 14 | 28 | 15.779 | 14.273 | -9.54% |
| | 18 | 36 | 17.797 | 17.064 | -4.12% |
| | 22 | 44 | 19.318 | 19.359 | 0.21% |
| 4KiB 100% Random Write | 1 | 2 | 1.079 | 1.079 | 0.07% |
| | 2 | 4 | 2.479 | 2.479 | -0.03% |
| | 6 | 12 | 7.126 | 7.156 | 0.42% |
| | 10 | 20 | 12.297 | 12.683 | 3.14% |
| | 14 | 28 | 13.096 | 12.425 | -5.13% |
| | 18 | 36 | 12.370 | 12.315 | -0.44% |
| | 22 | 44 | 12.670 | 12.667 | -0.03% |
| 4KiB 70% Random Read 30% Random Write | 1 | 2 | 1.338 | 1.359 | 1.57% |
| | 2 | 4 | 2.811 | 2.899 | 3.15% |
| | 6 | 12 | 8.104 | 7.949 | -1.91% |
| | 10 | 20 | 13.146 | 12.561 | -4.45% |
| | 14 | 28 | 13.874 | 12.808 | -7.69% |
| | 18 | 36 | 15.367 | 14.604 | -4.97% |
| | 22 | 44 | 16.325 | 16.170 | -0.95% |

Packed Ring performance impact

Selected test cases were re-run to show benefits of using Packed Rings as an option when configuring SPDK Vhost BLK controllers. For this, an optional parameter “--packed_ring” must be used when creating a SPDK Vhost BLK controller. Packed Ring feature requires QEMU 4.2.0 or later.

Table 10: Packed Ring performance impact on SPDK Vhost BLK controllers

| Workload | # of CPU cores | # of VMs | IOPS (millions) Split Ring | IOPS (millions) Packed Ring | Avg. Latency (usec) Split Ring | Avg. Latency (usec) Packed Ring | Packed Ring IOPS impact (%) | Packed Ring Avg. Latency impact (%) |
|---|----------------|----------|----------------------------|-----------------------------|--------------------------------|---------------------------------|-----------------------------|-------------------------------------|
| 4KiB 100% Random Read QD=64 | 1 | 2 | 14.62 | 15.29 | 175.23 | 167.00 | 4.61% | -4.69% |
| | 2 | 4 | 1.88 | 1.92 | 136.32 | 132.87 | 2.61% | -2.53% |
| | 6 | 12 | 14.54 | 14.85 | 249.83 | 245.23 | 2.11% | -1.84% |
| | 10 | 20 | 14.95 | 16.56 | 311.01 | 275.09 | 10.80% | -11.55% |
| | 14 | 28 | 18.69 | 18.73 | 300.68 | 302.52 | 0.24% | 0.61% |
| | 18 | 36 | 3.77 | 3.85 | 135.88 | 132.57 | 2.23% | -2.43% |
| | 22 | 44 | 9.03 | 9.45 | 171.22 | 160.85 | 4.63% | -6.05% |
| 4KiB 100% Random Write QD=64 | 1 | 2 | 12.89 | 13.12 | 199.74 | 197.30 | 1.77% | -1.23% |
| | 2 | 4 | 1.10 | 1.13 | 247.76 | 240.14 | 3.30% | -3.07% |
| | 6 | 12 | 13.17 | 13.46 | 275.77 | 266.28 | 2.22% | -3.44% |
| | 10 | 20 | 11.91 | 11.92 | 396.03 | 384.30 | 0.06% | -2.96% |
| | 14 | 28 | 11.30 | 11.92 | 502.83 | 474.33 | 5.55% | -5.67% |
| | 18 | 36 | 2.56 | 2.57 | 217.27 | 222.17 | 0.33% | 2.25% |
| | 22 | 44 | 7.22 | 7.35 | 212.58 | 221.15 | 1.79% | 4.03% |
| 4KiB 70% Random Read 30% Random Write QD=64 | 1 | 2 | 13.16 | 13.59 | 194.11 | 187.96 | 3.23% | -3.17% |
| | 2 | 4 | 1.50 | 1.48 | 173.54 | 169.19 | -1.24% | -2.51% |
| | 6 | 12 | 13.24 | 13.64 | 272.30 | 260.33 | 2.98% | -4.40% |
| | 10 | 20 | 13.79 | 14.47 | 333.09 | 317.52 | 4.94% | -4.67% |
| | 14 | 28 | 15.90 | 16.29 | 353.64 | 345.82 | 2.44% | -2.21% |
| | 18 | 36 | 2.91 | 2.97 | 174.71 | 174.14 | 1.72% | -0.33% |
| | 22 | 44 | 7.91 | 8.25 | 193.26 | 188.00 | 4.22% | -2.72% |

Conclusions

1. For SPDK Vhost-Scsi performance with split NVMe bdevs, we measured 1.71 million IOPS on one Vhost core for the 4KiB 100% Random Read workload. The single Vhost core IOPS for 4 KiB Random Write and 4KiB Random 70/30 Read/Write were 1.08 million and 1.34 million IOPS respectively. For all workloads, the IOPS scaled near linearly with addition of I/O processing cores up to 10 CPU cores. Peak performance was achieved at:
 - 22 CPU cores with 19.32 million IOPS for Random Read workload
 - 14 CPU cores with 13.10 million IOPS for Random Write workload
 - 22 CPU cores with 16.33 million IOPS for Random Read/Write workload
2. For SPDK Vhost-Scsi performance with Logical Volume backend devices, we measured 1.71 million IOPS on one Vhost core for the 4KiB 100% Random Read workload. The single Vhost core IOPS for 4 KiB Random Write and 4KiB Random 70/30 Read/Write were 1.08 million and 1.36 million IOPS respectively. For all workloads, the IOPS scaled with addition of I/O processing cores up to 10 CPU cores.

Peak performance was achieved at:

 - 22 CPU cores with 19.36 million IOPS for Random Read workload
 - 22 CPU cores with 12.67 million IOPS for Random Write workload
 - 22 CPU cores with 16.17 million IOPS for Random Read/Write workload
3. For SPDK Vhost-Blk with Logical Volume backend devices, we measured 1.79 million IOPS on one Vhost core for the 4KiB 100% Random Read workload. The single Vhost core IOPS for 4 KiB Random Write and 4KiB Random 70/30 Read/Write were 1.12 million and 1.46 million IOPS respectively. For all workloads, the IOPS scaled near linearly with addition of I/O processing cores up to 10 CPU cores. Peak performance was achieved at:
 - 22 CPU cores with 16.00 million IOPS for Random Read workload
 - 10 CPU cores with 12.58 million IOPS for Random Write workload
 - 22 CPU cores with 14.79 million IOPS for Random Read/Write workload
4. Using Logical Volumes has an impact of up to about 7-10% lower IOPS than when using Split NVMe block devices.
5. Using Packed Ring option instead of default Split Ring mode for SPDK Vhost BLK controllers results in minor performance improvement.

Test Case 2: Rate Limiting IOPS per VM

This test case was geared towards understanding how many VMs can be supported at a pre-defined Quality of Service of IOPS per Vhost device. Both read and write IOPS were rate limited for each Vhost device on each of the VMs and then VM density was compared between SPDK & the Linux Kernel. 25k IOPS per VM were chosen as the rate limiter using Linux cgroups2.

Each individual VM was running FIO with the following workloads:

- 4KiB 100% Random Read
- 4KiB 100% Random Write

The results in tables are average of 3 runs.

Table 11: Rate Limiting IOPS per VM test case configuration

| Item | Description |
|---|--|
| Test case | Test rate limiting IOPS/VM to 25000 IOPS |
| Test configuration | <p>FIO Version: fio-3.28</p> <p>VM Configuration:</p> <ul style="list-style-type: none"> • Common settings are described in the Virtual Machine Settings chapter. • Number of VMs: 24, 48, and 72 • Each VM has a single Vhost device which is one of equal partitions of NVMe drive. Total number of partitions depends on run test case. <ul style="list-style-type: none"> ○ For 24 VMs: 24xNVMe * 1 partition per NVMe = 24 partitions ○ For 48 VMs: 24xNVMe * 2 partitions per NVMe = 48 partitions ○ For 72 VMs: 24xNVMe * 3 partitions per NVMe = 72 partitions • Devices on VMs were throttled to run at a maximum of 25k IOPS (read or write) <p>SPDK Vhost target configuration:</p> <ul style="list-style-type: none"> • Test were run with both Vhost-Scsi and Vhost-Blk stacks. • The Vhost-Scsi stack was run with Split NVMe bdevs and Logical Volume bdevs. • The Vhost-Blk stack was run with Logical Volume bdevs. • Test were run with the Vhost target using 6 CPU cores (NUMA optimized). <p>Kernel Vhost-Scsi configuration:</p> <ul style="list-style-type: none"> • NUMA optimizations were not explored. • The Vhost kernel threads (single thread per vhost target) each is limited to using up to 6 CPU cores via cgroups (NUMA optimized). |
| FIO configuration run on each VM | [global] ioengine=libaio direct=1 |

| | |
|--|---|
| | <pre>rw=randrw rwmixread=100 (100% reads), 0 (100% writes) thread=1 norandommap=1 time_based=1 runtime=240s ramp_time=60s bs=4k iodepth=1 numjobs=1</pre> |
|--|---|

Test Case 2 Results

Table 12: 4KiB 100% Random Reads QD=1 rate limiting test results, 6 Vhost CPU cores

| # of VMs | Stack | Backend bdev | IOPS (k) | Avg Lat. (usec) |
|----------------|-------------|------------------|----------|-----------------|
| 24 VMs | SPDK-SCSI | Split NVMe | 291.51 | 82.10 |
| | SPDK-SCSI | Logical Volume | 290.72 | 82.27 |
| | SPDK-BLK | Logical Volume | 291.27 | 82.13 |
| | Kernel-SCSI | Partitioned NVMe | 248.35 | 96.36 |
| 48 VMs | SPDK-SCSI | Split NVMe | 577.38 | 82.88 |
| | SPDK-SCSI | Logical Volume | 575.70 | 83.12 |
| | SPDK-BLK | Logical Volume | 562.67 | 85.04 |
| | Kernel-SCSI | Partitioned NVMe | 451.90 | 105.94 |
| 72 VMs | SPDK-SCSI | Split NVMe | 811.56 | 88.31 |
| | SPDK-SCSI | Logical Volume | 812.75 | 88.20 |
| | SPDK-BLK | Logical Volume | 794.67 | 90.23 |
| | Kernel-SCSI | Partitioned NVMe | 376.55 | 190.68 |
| 96 VMs | SPDK-SCSI | Split NVMe | 1007.57 | 94.74 |
| | SPDK-SCSI | Logical Volume | 1028.90 | 92.77 |
| | SPDK-BLK | Logical Volume | 981.65 | 97.26 |
| | Kernel-SCSI | Partitioned NVMe | 363.17 | 263.81 |
| 106 VMs | SPDK-SCSI | Split NVMe | 1082.90 | 97.30 |
| | SPDK-SCSI | Logical Volume | 1102.45 | 95.58 |
| | SPDK-BLK | Logical Volume | 1033.20 | 102.02 |
| | Kernel-SCSI | Partitioned NVMe | 358.64 | 295.75 |

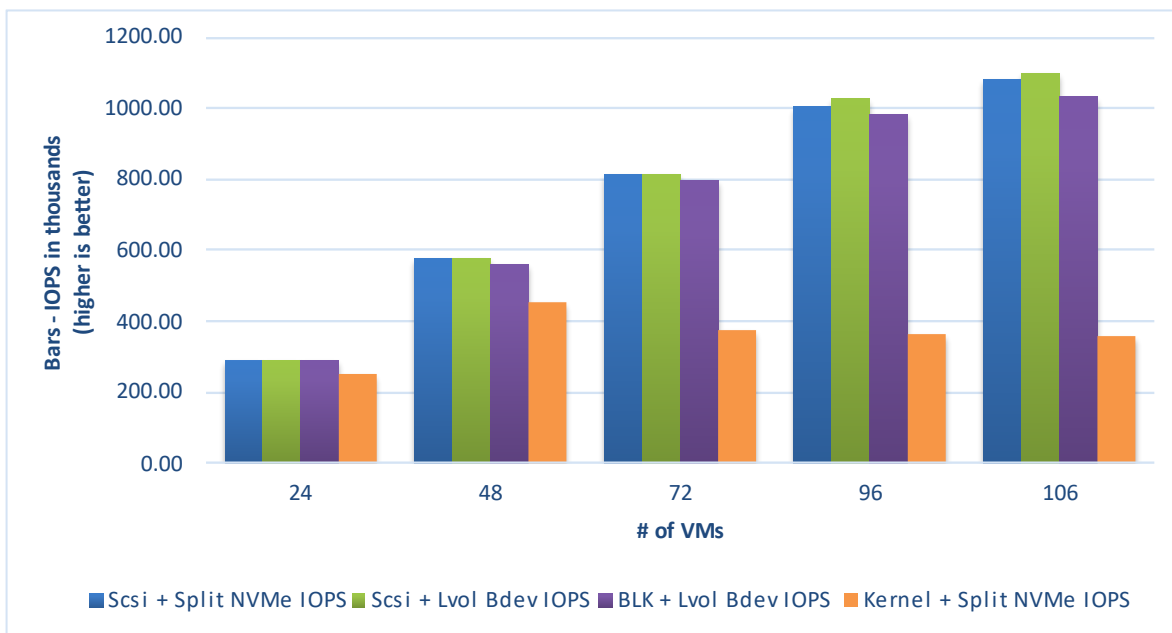


Figure 5: 4KiB 100% Random Reads IOPS, QD=1, throttling = 25k IOPS, 6 Vhost CPU cores

Table 13: 4KiB 100% Random Writes QD=1 rate limiting test results

| # of VMs | Stack | Backend bdev | IOPS (k) | Avg Lat. (usec) |
|----------------|-------------|------------------|----------|-----------------|
| 24 VMs | SPDK-SCSI | Split NVMe | 599.95 | 39.74 |
| | SPDK-SCSI | Logical Volume | 599.95 | 39.74 |
| | SPDK-BLK | Logical Volume | 599.99 | 39.74 |
| | Kernel-SCSI | Partitioned NVMe | 547.20 | 43.57 |
| 48 VMs | SPDK-SCSI | Split NVMe | 1199.90 | 39.74 |
| | SPDK-SCSI | Logical Volume | 1199.94 | 39.73 |
| | SPDK-BLK | Logical Volume | 1199.91 | 39.74 |
| | Kernel-SCSI | Partitioned NVMe | 411.58 | 116.59 |
| 72 VMs | SPDK-SCSI | Split NVMe | 1799.95 | 39.59 |
| | SPDK-SCSI | Logical Volume | 1799.84 | 39.59 |
| | SPDK-BLK | Logical Volume | 1750.57 | 40.75 |
| | Kernel-SCSI | Partitioned NVMe | 439.51 | 164.65 |
| 96 VMs | SPDK-SCSI | Split NVMe | 2211.69 | 42.84 |
| | SPDK-SCSI | Logical Volume | 2192.97 | 43.21 |
| | SPDK-BLK | Logical Volume | 1946.76 | 48.78 |
| | Kernel-SCSI | Partitioned NVMe | 327.71 | 295.40 |
| 106 VMs | SPDK-SCSI | Split NVMe | 2219.49 | 47.13 |
| | SPDK-SCSI | Logical Volume | 2086.46 | 50.21 |
| | SPDK-BLK | Logical Volume | 1924.99 | 54.35 |
| | Kernel-SCSI | Partitioned NVMe | 300.66 | 352.89 |

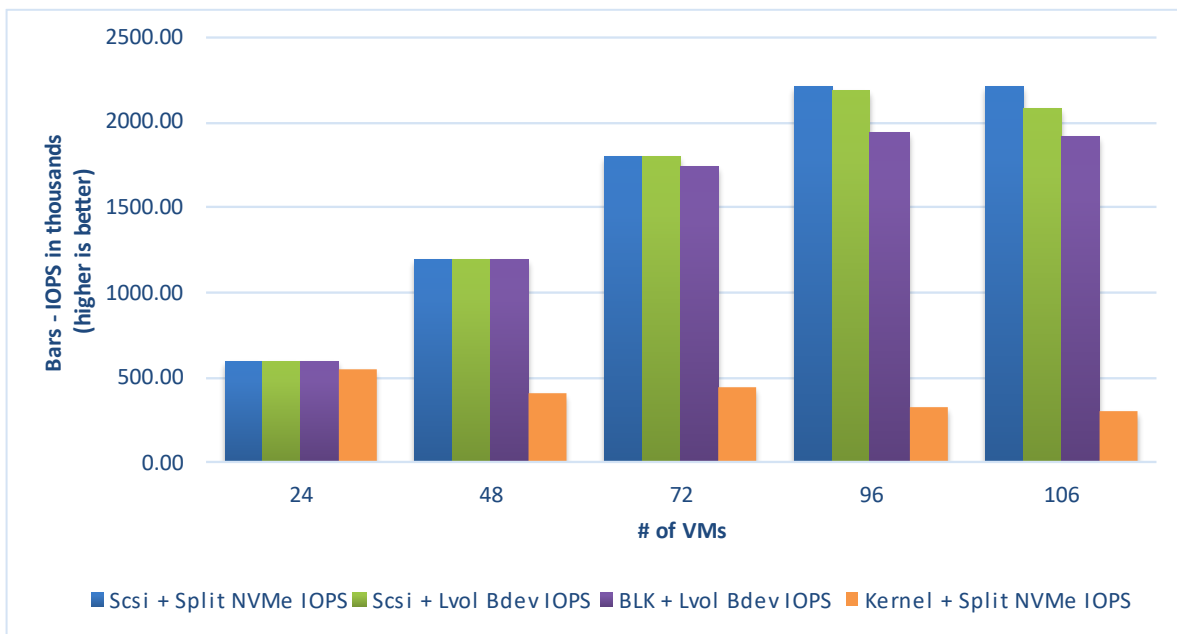


Figure 6: 4KiB 100% Random Writes IOPS, QD=1, throttling = 25k IOPS, 6 Vhost CPU cores

Conclusions

1. None of the tested Vhost solutions was able to serve 25,000 IOPS per VM for 4KiB Random Read workload.
2. Using 6 I/O processing cores, the SPDK Vhost serves 25,000 IOPS per VM to up to 72 VMs for 4 KiB Random Write workload.
3. Throughput and average latencies were up to 1.17x and 7.5x times better for Random Read and Random Write workloads respectively when using SPDK Vhost as compared to Kernel Vhost.
4. In the SPDK Vhost 24.01 performance report, we re-enabled CPU limitation on Kernel-Vhost using Linux cgroups. This rectification addressed a bug in our automation scripts that had previously overlooked such restrictions. By making this adjustment, we ensure a more equitable comparison, enabling us to accurately evaluate SPDK's performance advantages even when Kernel-Vhost operates under CPU constraints.

Test Case 3: Performance per NVMe drive

This test case was performed to understand performance and efficiency of the Vhost-Scsi and Vhost-Blk process using SPDK vs. Linux Kernel with a single NVMe drive on 2 VMs. Each VM had a single Vhost device which is one of two equal partitions of an NVMe drive. Results in the table represent performance (IOPS, avg. latency & CPU %) seen from the VM. The VM was running FIO with the following workloads:

- 4KiB 100% Random Read
- 4KiB 100% Random Write
- 4KiB Random 70% Read 30% Write

The results in tables are average of 3 runs.

Table 14: Performance per NVMe drive test case configuration

| Item | Description |
|---------------------------|---|
| Test case | Test SPDK Vhost target I/O core scaling performance |
| Test configuration | <p>FIO Version: fio-3.28</p> <p>VM Configuration:</p> <ul style="list-style-type: none"> • Common settings are described in the Virtual Machine Settings chapter. • 2 VMs were tested. • Each VM had a single Vhost device which was one of two equal partitions of a single NVMe drive. <p>SPDK Vhost target configuration:</p> <ul style="list-style-type: none"> • The SPDK Vhost process was run on a single physical CPU core. • The Vhost-Scsi stack was run with Split NVMe bdevs and Logical Volume bdevs. • The Vhost-Blk stack was run with Logical Volume bdevs. <p>Kernel Vhost target configuration:</p> <ul style="list-style-type: none"> • The Vhost kernel threads (single thread per vhost target) each is limited to using up 1 CPU cores via cgroups (NUMA optimized). |
| FIO configuration | <pre>[global] ioengine=libaio direct=1 rw=randrw rwmixread=100 (100% reads), 70 (70% reads, 30% writes), 0 (100% writes) thread=1 norandommap=1 time_based=1 runtime=240s ramp_time=60s bs=4k iodepth= {1, 8, 32, 64, 128, 192} numjobs=1</pre> |

Test Case 3 results

SPDK Vhost-Scsi

Table 15: Performance per NVMe drive IOPS and latency results, SPDK SCSI stack

| Access pattern | Backend | QD | Throughput (IOPS k) | Avg. latency (usec) |
|-------------------------|------------|-----|---------------------|---------------------|
| 4KiB 100% Random Reads | Split NVMe | 1 | 24.26 | 82.24 |
| 4KiB 100% Random Reads | Split NVMe | 8 | 185.64 | 85.93 |
| 4KiB 100% Random Reads | Split NVMe | 32 | 618.94 | 103.10 |
| 4KiB 100% Random Reads | Split NVMe | 64 | 725.48 | 176.03 |
| 4KiB 100% Random Reads | Split NVMe | 128 | 732.28 | 348.71 |
| 4KiB 100% Random Reads | Split NVMe | 192 | 726.66 | 530.41 |
| 4KiB 100% Random Reads | Lvol | 1 | 24.17 | 82.39 |
| 4KiB 100% Random Reads | Lvol | 8 | 185.33 | 86.08 |
| 4KiB 100% Random Reads | Lvol | 32 | 616.76 | 103.30 |
| 4KiB 100% Random Reads | Lvol | 64 | 736.53 | 173.37 |
| 4KiB 100% Random Reads | Lvol | 128 | 740.43 | 344.96 |
| 4KiB 100% Random Reads | Lvol | 192 | 735.56 | 522.28 |
| 4KiB 100% Random Writes | Split NVMe | 1 | 136.78 | 14.34 |
| 4KiB 100% Random Writes | Split NVMe | 8 | 520.33 | 31.93 |
| 4KiB 100% Random Writes | Split NVMe | 32 | 641.88 | 99.86 |
| 4KiB 100% Random Writes | Split NVMe | 64 | 636.30 | 200.78 |
| 4KiB 100% Random Writes | Split NVMe | 128 | 638.65 | 399.33 |
| 4KiB 100% Random Writes | Split NVMe | 192 | 661.79 | 581.04 |
| 4KiB 100% Random Writes | Lvol | 1 | 136.15 | 14.43 |
| 4KiB 100% Random Writes | Lvol | 8 | 519.98 | 31.91 |
| 4KiB 100% Random Writes | Lvol | 32 | 654.71 | 97.58 |
| 4KiB 100% Random Writes | Lvol | 64 | 638.98 | 200.50 |
| 4KiB 100% Random Writes | Lvol | 128 | 664.42 | 385.09 |
| 4KiB 100% Random Writes | Lvol | 192 | 666.06 | 575.04 |
| 4KiB 70%/30% Random R/W | Split NVMe | 1 | 33.40 | 59.91 |
| 4KiB 70%/30% Random R/W | Split NVMe | 8 | 223.52 | 71.47 |
| 4KiB 70%/30% Random R/W | Split NVMe | 32 | 568.05 | 112.43 |
| 4KiB 70%/30% Random R/W | Split NVMe | 64 | 653.69 | 195.72 |
| 4KiB 70%/30% Random R/W | Split NVMe | 128 | 681.85 | 375.05 |
| 4KiB 70%/30% Random R/W | Split NVMe | 192 | 689.89 | 556.13 |
| 4KiB 70%/30% Random R/W | Lvol | 1 | 33.12 | 60.50 |
| 4KiB 70%/30% Random R/W | Lvol | 8 | 225.54 | 70.78 |
| 4KiB 70%/30% Random R/W | Lvol | 32 | 565.30 | 113.10 |
| 4KiB 70%/30% Random R/W | Lvol | 64 | 668.73 | 191.15 |
| 4KiB 70%/30% Random R/W | Lvol | 128 | 690.81 | 370.43 |
| 4KiB 70%/30% Random R/W | Lvol | 192 | 681.98 | 562.13 |

SPDK Vhost-Blk

Table 16: Performance per NVMe drive IOPS and latency results. SPDK BLK stack

| Access pattern | Backend | QD | Throughput (IOPS k) | Avg. latency (usec) |
|-------------------------|---------|-----|---------------------|---------------------|
| 4KiB 100% Random Reads | Lvol | 1 | 24.26 | 82.12 |
| 4KiB 100% Random Reads | Lvol | 8 | 185.84 | 85.90 |
| 4KiB 100% Random Reads | Lvol | 32 | 629.09 | 101.42 |
| 4KiB 100% Random Reads | Lvol | 64 | 833.55 | 152.99 |
| 4KiB 100% Random Reads | Lvol | 128 | 831.88 | 307.29 |
| 4KiB 100% Random Reads | Lvol | 192 | 836.16 | 459.85 |
| 4KiB 100% Random Writes | Lvol | 1 | 138.77 | 14.15 |
| 4KiB 100% Random Writes | Lvol | 8 | 532.05 | 31.56 |
| 4KiB 100% Random Writes | Lvol | 32 | 691.13 | 92.23 |
| 4KiB 100% Random Writes | Lvol | 64 | 690.35 | 185.28 |
| 4KiB 100% Random Writes | Lvol | 128 | 721.92 | 354.67 |
| 4KiB 100% Random Writes | Lvol | 192 | 693.20 | 552.68 |
| 4KiB 70%/30% Random R/W | Lvol | 1 | 32.78 | 60.55 |
| 4KiB 70%/30% Random R/W | Lvol | 8 | 219.47 | 72.82 |
| 4KiB 70%/30% Random R/W | Lvol | 32 | 570.95 | 111.70 |
| 4KiB 70%/30% Random R/W | Lvol | 64 | 691.07 | 184.53 |
| 4KiB 70%/30% Random R/W | Lvol | 128 | 807.09 | 317.28 |
| 4KiB 70%/30% Random R/W | Lvol | 192 | 770.21 | 499.93 |

Kernel Vhost-Scsi

Table 17: Performance per NVMe drive IOPS and latency results. Kernel Vhost-Scsi

| Access pattern | Backend | QD | Throughput (IOPS k) | Avg. latency (usec) |
|-------------------------|---------|-----|---------------------|---------------------|
| 4KiB 100% Random Reads | NVMe | 1 | 21.68 | 92.00 |
| 4KiB 100% Random Reads | NVMe | 8 | 150.25 | 106.19 |
| 4KiB 100% Random Reads | NVMe | 32 | 319.37 | 200.24 |
| 4KiB 100% Random Reads | NVMe | 64 | 356.63 | 358.92 |
| 4KiB 100% Random Reads | NVMe | 128 | 413.47 | 618.23 |
| 4KiB 100% Random Reads | NVMe | 192 | 413.72 | 927.02 |
| 4KiB 100% Random Writes | NVMe | 1 | 75.91 | 26.07 |
| 4KiB 100% Random Writes | NVMe | 8 | 264.31 | 60.65 |
| 4KiB 100% Random Writes | NVMe | 32 | 339.36 | 188.66 |
| 4KiB 100% Random Writes | NVMe | 64 | 317.47 | 402.76 |
| 4KiB 100% Random Writes | NVMe | 128 | 290.65 | 880.14 |
| 4KiB 100% Random Writes | NVMe | 192 | 311.58 | 1231.48 |
| 4KiB 70%/30% Random R/W | NVMe | 1 | 27.77 | 70.30 |
| 4KiB 70%/30% Random R/W | NVMe | 8 | 170.32 | 93.18 |
| 4KiB 70%/30% Random R/W | NVMe | 32 | 302.79 | 211.30 |
| 4KiB 70%/30% Random R/W | NVMe | 64 | 308.17 | 415.00 |
| 4KiB 70%/30% Random R/W | NVMe | 128 | 322.56 | 795.03 |
| 4KiB 70%/30% Random R/W | NVMe | 192 | 320.77 | 1197.31 |

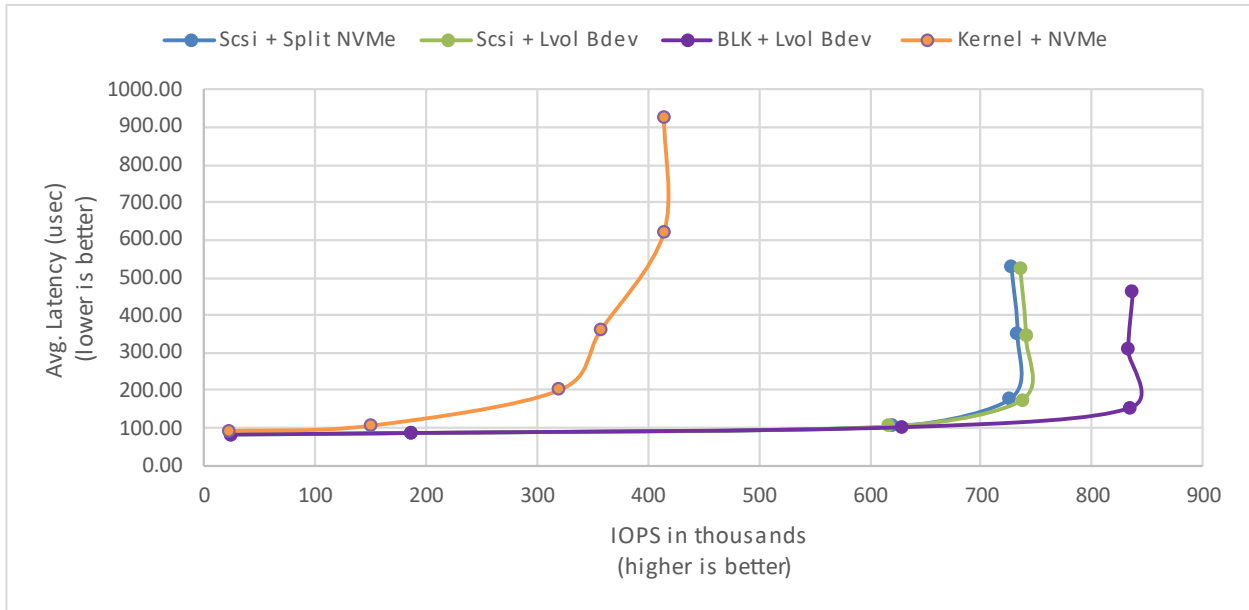


Figure 7: 4KiB 100% Random Reads IOPS and latency

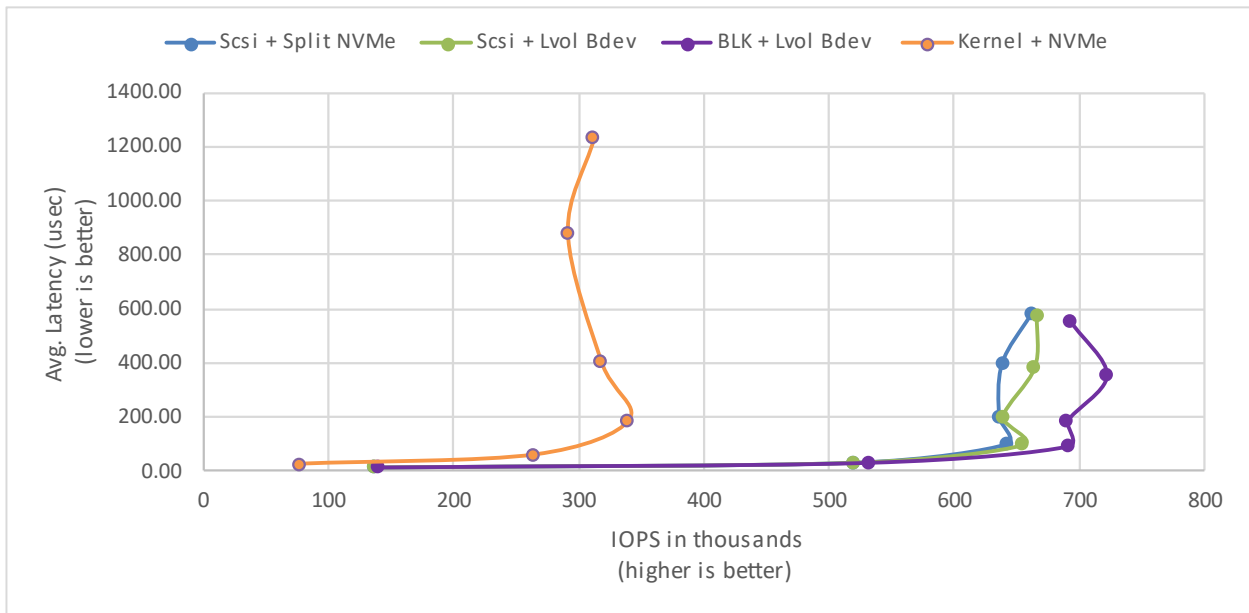


Figure 8: 4KiB 100% Random Writes IOPS and latency

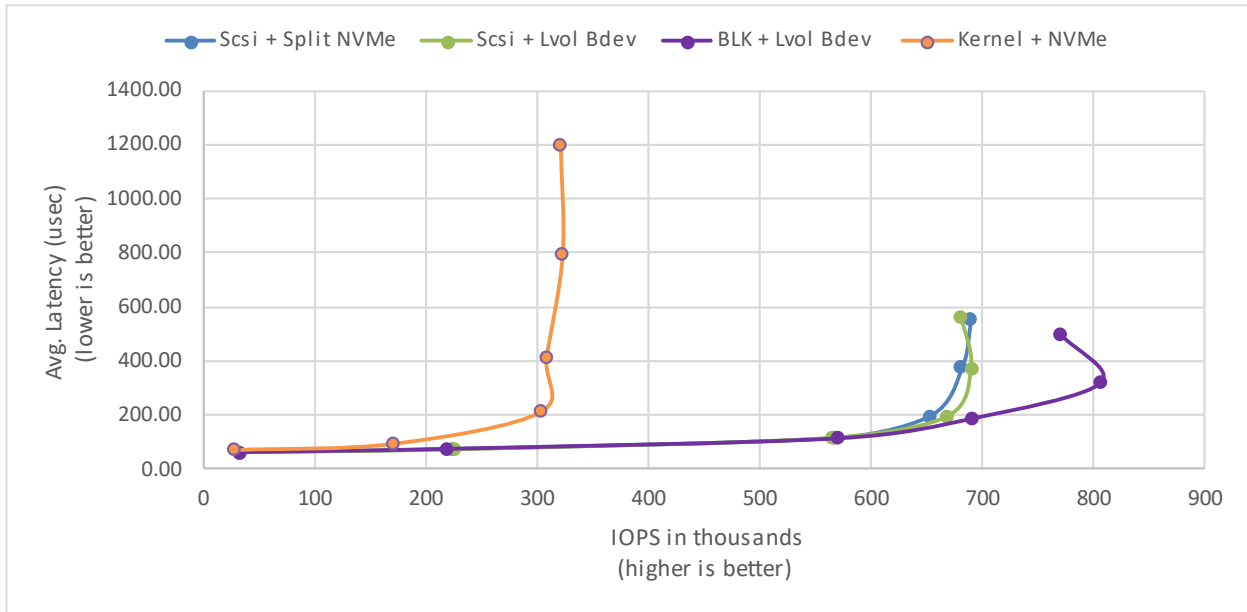


Figure 9: 4KiB 70%/30% Random Read/Write IOPS and latency

Conclusions

1. SPDK Vhost-Scsi with NVMe Split bdevs has lower latency and higher throughput than Kernel Vhost-Scsi in all of workload / queue depth combinations.
2. SPDK Vhost-Scsi with split NVMe were very similar and offered comparable IOPS and latency in most cases to SPDK Vhost-Scsi with Lvol Bdev
3. SPDK Vhost-BLK with lvol bdev delivered the best results for high QD values.

Summary

This report compared performance results while running Vhost-Scsi using traditional interrupt-driven kernel Vhost-Scsi against the accelerated polled-mode driven SPDK implementation. Various local ephemeral configurations were demonstrated, including rate limiting IOPS, performance per VM and maximum performance from an underlying system when comparing kernel vs. SPDK Vhost-Scsi target implementations.

In addition, performance impacts of using SPDK Logical Volume Bdevs and the SPDK Vhost-Blk stack were presented.

This report provided information regarding methodologies and practices while benchmarking Vhost-Scsi and Vhost-Blk using both SPDK and the Linux Kernel. It should be noted that the performance data showcased in this report is based on specific hardware and software configurations and that performance results may vary depending on different hardware and software configurations.

List of Tables

| | |
|--|----|
| Table 1: Hardware setup configuration | 4 |
| Table 2: Test platform BIOS settings | 5 |
| Table 3: Test System NVMe storage setup | 5 |
| Table 4: Guest VM configuration | 5 |
| Table 5: SPDK Vhost Core Scaling test configuration | 9 |
| Table 6: SPDK Vhost core scaling results, 4KiB 100% Random Reads IOPS, QD=64 | 11 |
| Table 7: SPDK Vhost core scaling results, 4KiB 100% Random Write IOPS, QD=64 | 12 |
| Table 8: SPDK Vhost core scaling results, 4KiB Random 70% Read 30% Write IOPS, QD=64 | 13 |
| Table 9: Logical Volumes performance impact for SPDK Vhost SCSI | 14 |
| Table 10: Packed Ring performance impact on SPDK Vhost BLK controllers | 15 |
| Table 11: Rate Limiting IOPS per VM test case configuration | 17 |
| Table 12: 4KiB 100% Random Reads QD=1 rate limiting test results, 6 Vhost CPU cores | 19 |
| Table 13: 4KiB 100% Random Writes QD=1 rate limiting test results | 20 |
| Table 14: Performance per NVMe drive test case configuration | 22 |
| Table 15: Performance per NVMe drive IOPS and latency results, SPDK SCSI stack | 23 |
| Table 16: Performance per NVMe drive IOPS and latency results. SPDK BLK stack | 24 |
| Table 17: Performance per NVMe drive IOPS and latency results. Kernel Vhost-Scsi | 24 |

List of Figures

| | |
|---|-----------|
| <i>Figure 1: SPDK Vhost-scsi architecture</i> | <i>7</i> |
| <i>Figure 2: Comparison of performance between various SPDK Vhost stack-bdev combinations for 4KiB Random Read QD=64 workload</i> | <i>11</i> |
| <i>Figure 3: Comparison of performance between various SPDK Vhost stack-bdev combinations for 4KiB Random Write QD=64 workload</i> | <i>12</i> |
| <i>Figure 4: Comparison of performance between various SPDK Vhost stack-bdev combinations for 4KiB Random 70% Read 30% Write QD=64 workload</i> | <i>13</i> |
| <i>Figure 5: 4KiB 100% Random Reads IOPS, QD=1, throttling = 25k IOPS, 6 Vhost CPU cores</i> | <i>19</i> |
| <i>Figure 6: 4KiB 100% Random Writes IOPS, QD=1, throttling = 25k IOPS, 6 Vhost CPU cores</i> | <i>20</i> |
| <i>Figure 7: 4KiB 100% Random Reads IOPS and latency</i> | <i>25</i> |
| <i>Figure 8: 4KiB 100% Random Writes IOPS and latency</i> | <i>25</i> |
| <i>Figure 9: 4KiB 70%/30% Random Read/Write IOPS and latency</i> | <i>26</i> |

Notices & Disclaimers

Performance varies by use configuration and other factors. Learn more at [Intel Performance Index](#).

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

§