Performance Characterization of NVMe-oF in vSphere 7.0 U1

Performance Study - November 23, 2020

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Table of Contents

Executive Summary

NVMe® continues to become increasingly popular because of its low latency and high throughput. VMware vSphere® 7.0 introduces support for NVMe over Fabrics (NVMe-oF™), a protocol specification that connects hosts to high-speed flash storage via network fabrics using the NVMe protocol. The NVMe-oF fabrics that vSphere 7.0 supports include Fibre Channel (FC-NVMe) and RDMA (RoCE v2).

The purpose of this paper is to compare the performance of the legacy Fibre Channel Protocol (SCSI FCP) to FC-NVMe on vSphere 7.0 U1. The same host bus adapter (HBA) and storage area network (SAN) are used in all benchmark runs. The benchmark results show that FC-NVMe consistently outperforms SCSI FCP in virtualized environments, providing higher throughput and lower latency. These results provide a compelling case for customers to upgrade their existing environments to vSphere 7.0 and gain the benefits of NVMe-oF.

Introduction

The most prevalent protocol used to connect ESXi hosts to SAN storage arrays is Fibre Channel [1]. It has been around for decades, and one of the reasons for its continued dominance is that it was built specifically for storage reliability, scalability, and high performance [2]. SCSI standards date back to the 1980s and thus were geared towards rotating hard disk drives. Over time, the SCSI command set has become relatively bloated, and the traditional maximum I/O queue depths of 32 or 64 are insufficient for flash storage. The NVMe protocol is a NUMA-optimized protocol and was developed to address many shortcomings of SCSI. It supports tens of thousands of parallel command queues which can be used as submission and completion queues on different CPUs; this helps the implementation to be lockless. It has a minimal number of streamlined commands just for flash technology [3]. As enterprise all-flash arrays (AFA) become more prevalent in the data center, the traditional SCSI Fibre Channel Protocol can become a potential performance bottleneck for the NVMe-based devices and infrastructure available today and coming in the near future.

The NVMe base specification, however, only covered direct-attached storage; that is, SSDs attached directly to the PCIe bus of the host. The NVMe-oF specification was developed to take advantage of the benefits of the NVMe protocol over networked fabrics (that is, connecting to an external SAN). vSphere 7.0 supports both NVMe over Fibre Channel and NVMe over RDMA, but this technical paper covers only performance results using a Fibre Channel SAN array.

The key advantage of Fibre Channel infrastructure is that both SCSI FCP and FC-NVMe can be run on the same fabric, allowing for easy interoperability and data migration. One caveat is that the FC switches, HBAs, and/or storage arrays may need to be upgraded if they are not FC-NVMe capable; in some cases, this can be accomplished through a simple firmware (software) flash, avoiding any investment in additional hardware.

The goal of this paper is to demonstrate the best possible increase in IOPS and decrease in latency when moving from SCSI FCP to FC-NVMe technology under ideal conditions. This paper details the hardware and software test environment used for the performance tests, compares the SCSI FCP to the FC-NVMe results, and summarizes the key findings based on the benchmark results. There were two separate sets of benchmarks run:

1. Microbenchmarks obtained using a synthetic workload generator (fio) with varying combinations of reads/writes, sequential/random mixes, and I/O sizes.
2. A real-world database benchmark:Microsoft® Cloud Database Benchmark (CDB), which stressed vSphere virtual machines running Microsoft SQL Server.

Test Environment

Host Hardware and Software

Configuration used for tests:

- Dell PowerEdge™ R740xd host, which contained two Intel® Xeon® Gold 6154 processors with 36 total cores running at 3.00 GHz with 190 GB of RAM
- VMware ESXi™ 7.0 Update 1 (figure 1).

Connection to the Fibre Channel storage array:

- Dual-port Broadcom Emulex® LPe35002-M2 FC HBA , with the default driver version 12.6.278.2

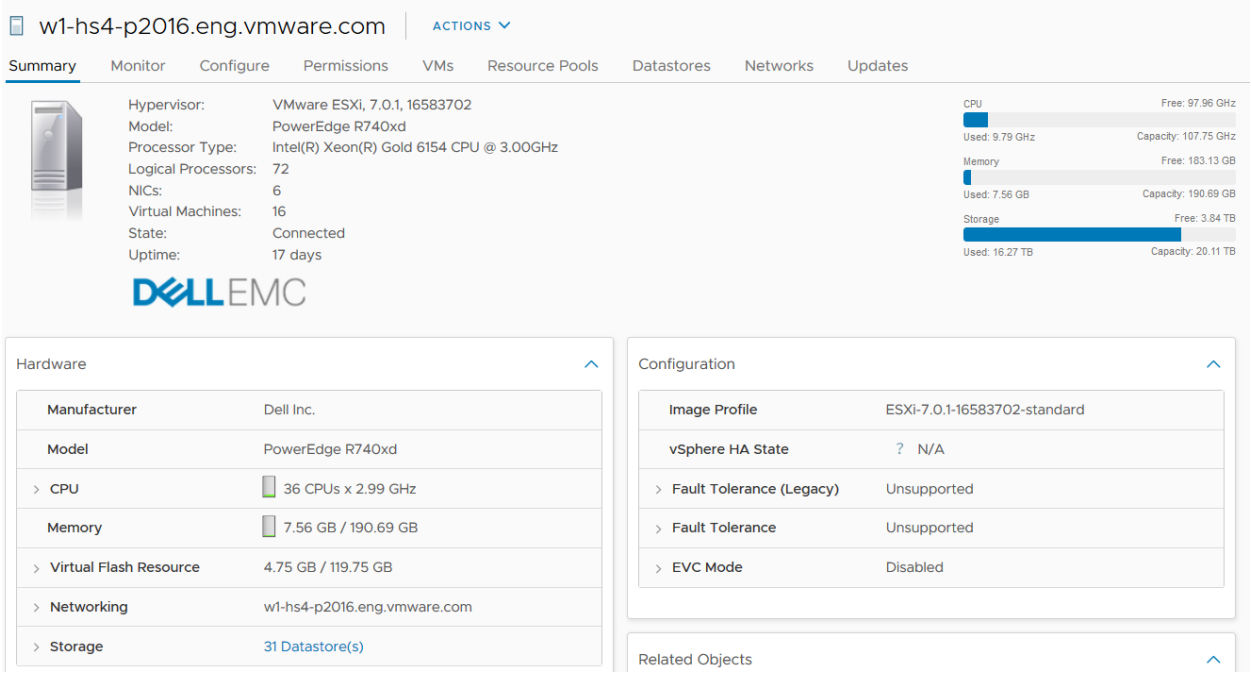

Figure 1. Screenshot of the host used

Storage Hardware

- Dell EMC Connectrix™ DS-6620B Fibre Channel switch was used to connect the host to the SAN.
- NetApp® AFF A800 array was used for both the SCSI FCP and FC-NVMe storage results. It consisted of 24 1.75 TB NVMe SSDs, as shown in figure 2.

```
wl-hs4-fas-a800-01::> storage disk show
                  Usable           Disk    Container   Container
Disk                Size Shelf Bay Type    Type        Name      Owner
---------------- ---------- ----- --- ------- ----------- --------- --------

Info: This cluster has partitioned disks. To get a complete list of spare disk capacity use "storage aggregate show-spare-disks".
1.0.0             1.75TB     0   0 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.1             1.75TB     0   1 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.2             1.75TB     0   2 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.6             1.75TB     0   6 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.7             1.75TB     0   7 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.8             1.75TB     0   8 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.12            1.75TB     0  12 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.13            1.75TB     0  13 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.14            1.75TB     0  14 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.18            1.75TB     0  18 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_01, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.19            1.75TB     0  19 SSD-NVM shared      wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1 wl-hs4-fas-a800-01-01
1.0.20            1.75TB     0  20 SSD-NVM shared      -         wl-hs4-fas-a800-01-01
1.0.24            1.75TB     0  24 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.25            1.75TB     0  25 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.26            1.75TB     0  26 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.30            1.75TB     0  30 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.31            1.75TB     0  31 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.32            1.75TB     0  32 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.36            1.75TB     0  36 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.37            1.75TB     0  37 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.38            1.75TB     0  38 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.42            1.75TB     0  42 SSD-NVM shared      aggr0_wl_hs4_fas_a800_01_02, wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1
1.0.43            1.75TB     0  43 SSD-NVM shared      wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1 wl-hs4-fas-a800-01-02
1.0.44            1.75TB     0  44 SSD-NVM shared      wl_hs4_fas_a800_01_01_NVME_SSD_1, wl_hs4_fas_a800_01_02_NVME_SSD_1 wl-hs4-fas-a800-01-02
24 entries were displayed.

wl-hs4-fas-a800-01::>
```

Figure 2. Screenshot of storage array disk configuration

Connectivity Diagram

Figure 3 shows a logical diagram of the key hardware components used in our test environment.

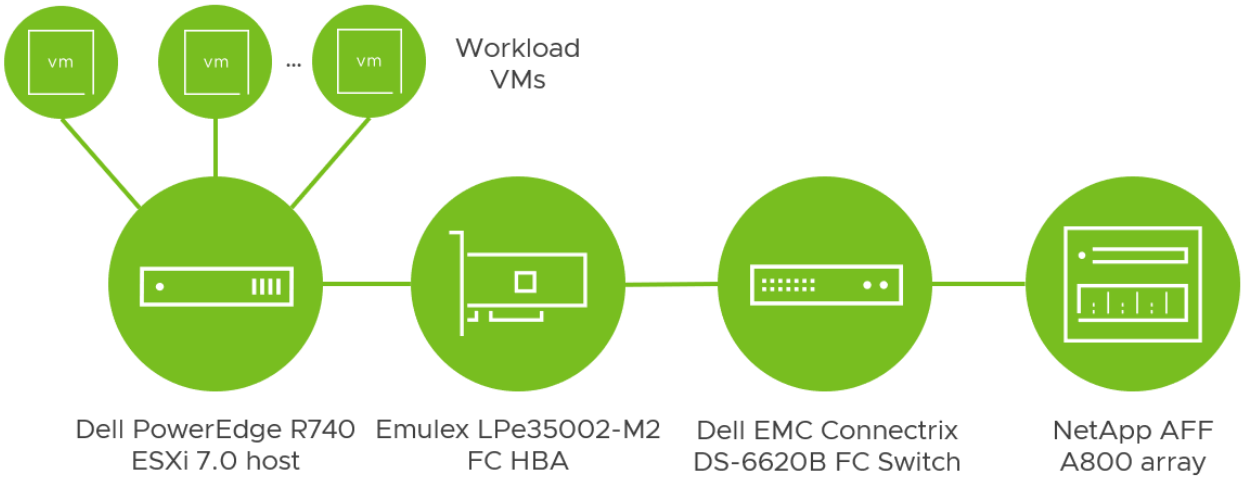

Figure 3. Logical diagram of hardware components

Benchmarks

We used two benchmark workloads to measure performance for this study: fio (to generate different I/O sizes and patterns) and CDB (for OLTP/SQL Server performance).

- Fio (Flexible I/O Tester) was written by Jens Axboe to test the Linux I/O subsystem and schedulers, either to find/reproduce bugs or to test performance [4]. Any mixture of reads and writes, sequential vs. random I/O, and threads or processes can be easily specified.
- CDB (Cloud Database Benchmark) is a database schema and workload mix designed by Microsoft. The benchmark is designed with cloud computing in mind, but the database schema, data population, and transactions have been designed to be broadly representative of the basic elements most commonly used in OLTP workloads. The benchmark driver reports throughput in terms of transactions per second. Since the benchmark is relatively new, the workload was designed to simulate heavier loads on the database with each simulated user/thread. We were first introduced to this workload during our project to measure the performance of SQL Server on Linux using vSphere [5].

Virtual Machine Configuration

For the fio microbenchmark tests, four Ubuntu 18.04 virtual machines were used. Each VM consisted of six 1 GB *thick eager zeroed* (EZT) disks per namespace (4 namespaces in total for four VMs). There were a total of 120 outstanding IOs (OIO), 30 per device, and four PVSCSI storage adapters.

For the SQL Server benchmark tests, eight SQL Server database virtual machines were installed with Red Hat Enterprise Linux 8.1 and SQL Server 2019. Each VM was configured with 16 virtual CPUs, 16 GB of RAM, a 1 TB thick eager zeroed disk on a dedicated VMFS 6 datastore that held the OS, database, and logs, a PVSCSI storage adapter, and a VMXNET3 network adapter. We followed the Microsoft quickstart guide to install SQL Server using the RHEL 8 repository as well as their best practices for configuring Linux and SQL Server to improve performance [6].

The CDB virtual machines (which generated the SQL Server benchmark load) were installed with Windows Server 2019 as the guest operating system (OS). Each VM was configured with 16 virtual CPUs, 32 GB of RAM, a 200 GB thin provisioned disk on a VMFS 6 datastore for the OS, a PVSCSI storage adapter, and a VMXNET3 network adapter.

Performance Results

Microbenchmark (fio) Results

The fio microbenchmark runs consisted of random and sequential patterns, reads, and writes for various IO sizes from 4 KB to 256 KB, comparing the IOPS and latency between FC-NVMe and SCSI FCP targets on a NetApp A800 array with Broadcom cards on the ESXi host. The FC-NVMe microbenchmark runs had the virtual disks placed on four NVMe namespaces with a single controller, and the SCSI FCP microbenchmark runs had virtual disks placed on four different SCSI LUNs.

As seen in Figures 4 and 6, the IOPS from FC-NVMe runs are double the IOPS from the SCSI FCP target runs for all the IO sizes. As seen in the latency plots in Figures 5 and 7, the latencies from the FC-NVMe runs are clearly lower than latencies from SCSI FCP microbenchmark runs. Note that all these graphs are plotted on log scale and hence for example, a straight line indicates that the IOPS is doubling as the IO size decreases by half. Please note that these results were obtained using a microbenchmark like fio and with configurations that follow the best practices mentioned below; performance with other application workloads may vary.

Figure 4. IOPS - microbenchmark Random IO pattern runs

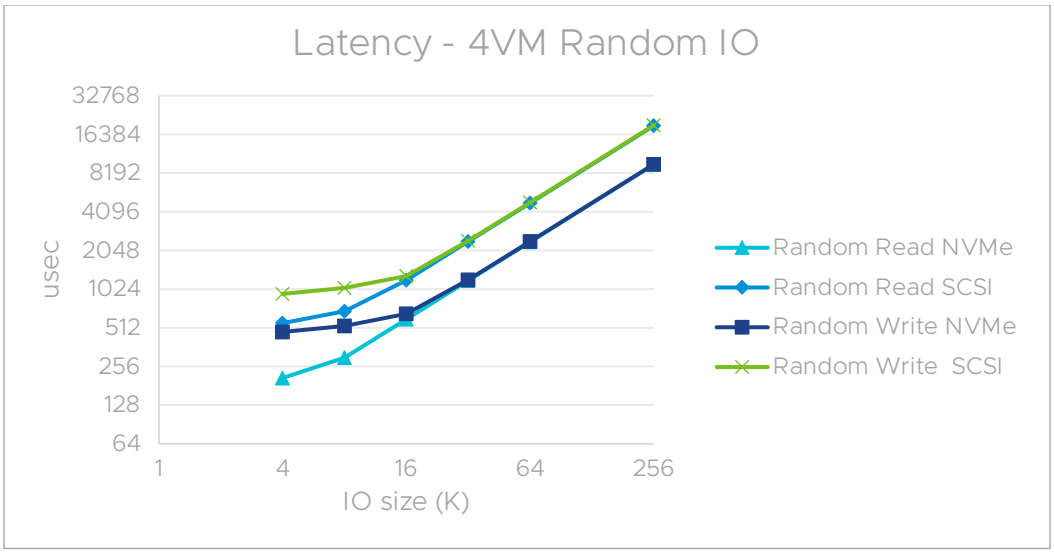

Figure 5. Latency - microbenchmark Random IO pattern runs

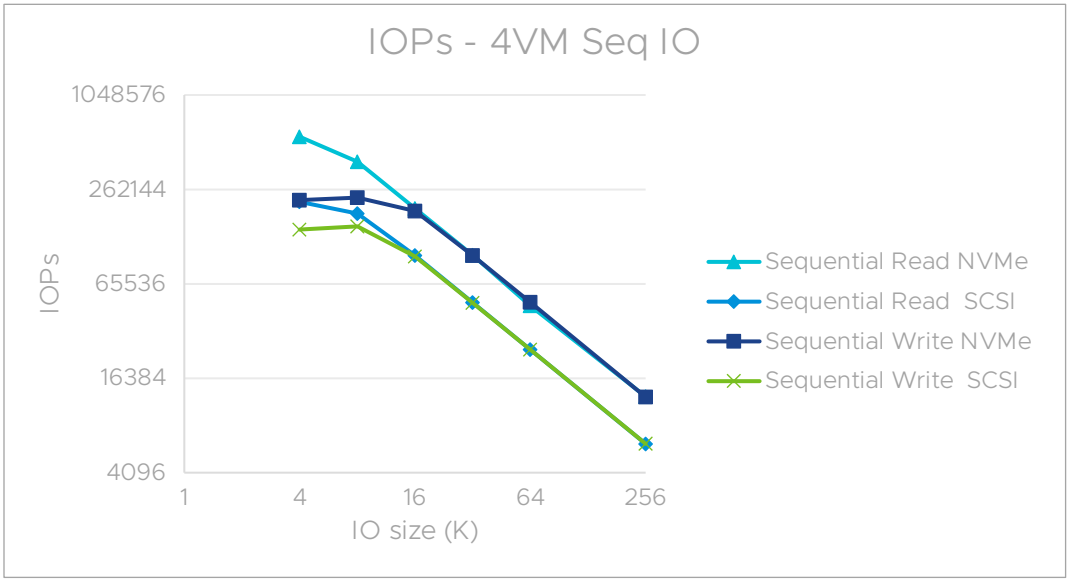

Figure 6. IOPS - microbenchmark Sequential IO pattern runs

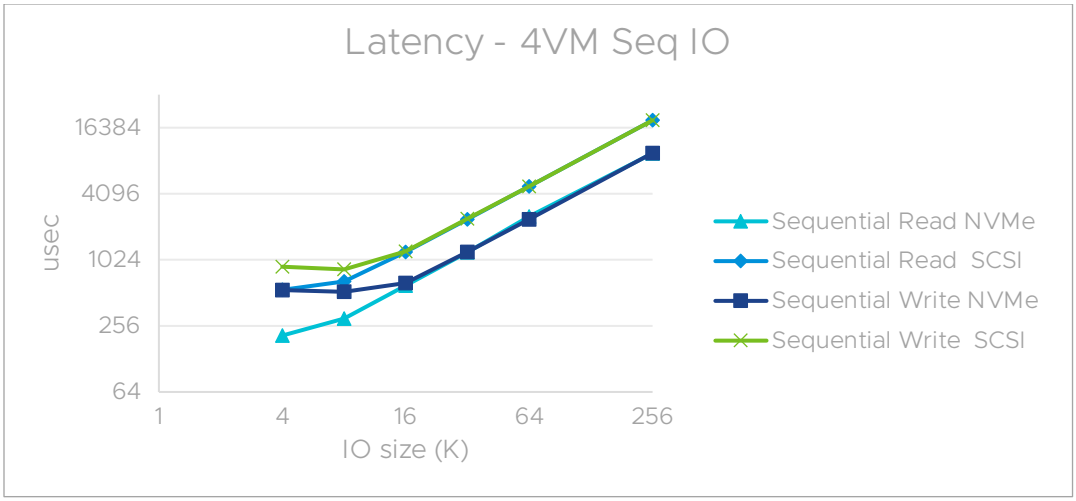

Figure 7. Latency - microbenchmark Sequential IO pattern runs

Some of the best practices followed in these runs in order to ensure high performance, apart from ensuring enough resources like vCPUs and memory are configured in the VMs, are:

- Spread out the virtual disks in VMs across as many virtual controllers as possible
- Ensure that the VMware high performance plug-in (HPP) module claims the NVMe disks
- Ensure that the VMs and vNUMA configurations are sized appropriately [7]
- Ensure that the host BIOS power settings are configured for performance and not for performance-power tradeoff [8]

SQL Server (CDB) Results

These tests involved driving a heavy database workload using the CDB benchmark using 1, 2, 4, and 8 SQL Server VMs on the ESXi host, both with legacy SCSI FCP and the newer FC-NVMe protocol.

The first chart that follows represents the primary CDB metric for measuring database performance: transactions per second (Txns/sec). FC-NVMe is higher in every case and is 85% higher, performing with 2 VMs. Note also that SCSI FCP never quite reached 10,000 Txns/sec, while FC-NVMe was able to surpass 11,000 Txns/sec with both 4 and 8 VMs.

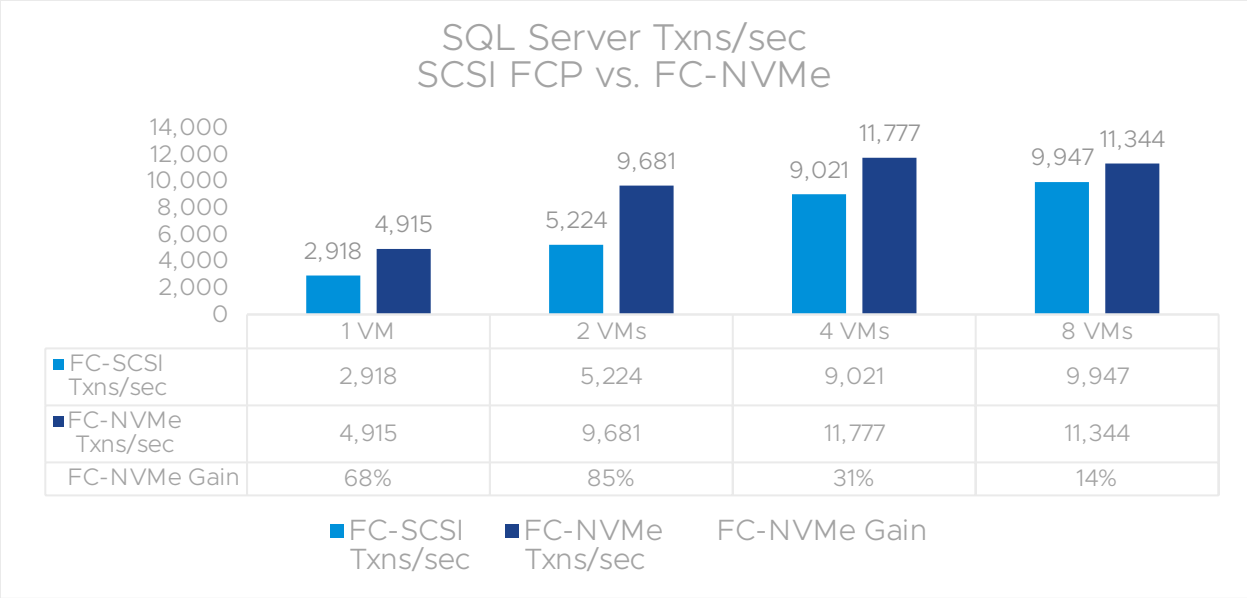

Figure 8. SQL Server Txns/sec for SCSI FCP vs. FC-NVMe

We also measured the disk IOPS (input/output operations per second) between the host and the SAN during the SQL Server benchmarks, which showed similar improvements, as the chart below shows.

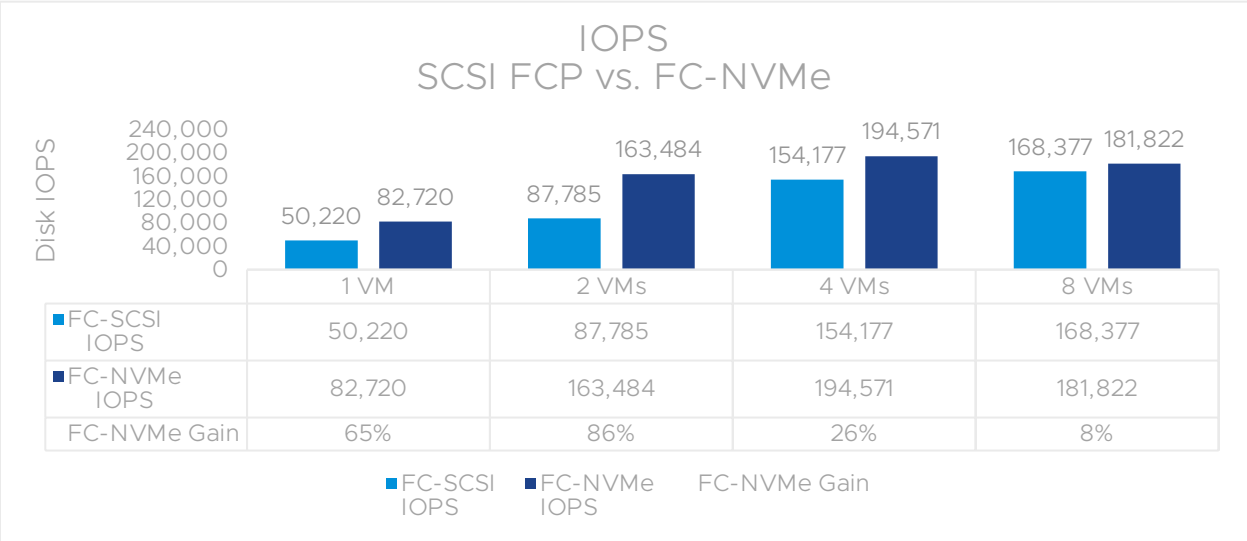

Figure 9. SQL Server IOPS for SCSI FCP vs. FC-NVMe

The slight drop in performance from 4 VMs to 8 VMs for FC-NVMe was due to CPU and memory utilization on the host, not a bottleneck on the storage subsystem side.

Conclusion

The benchmark results in this performance study show that customers who upgrade their legacy SAN infrastructure from SCSI FCP to FC-NVMe can achieve noticeably higher storage performance using vSphere 7.0. The microbenchmark results showed FC-NVMe was able to achieve twice the IOPS with lower latencies compared to SCSI FCP for every IO size tested. The SQL Server results show that almost the same amount of database throughput could be achieved using only half the number of SQL Server database VMs on FC-NVMe versus SCSI FCP. Performance with other application workloads may vary depending on the deployment configuration.

References

[1] NetApp. (2020) What Is a Storage Area Network? https://www.netapp.com/data-storage/what-is-san-storage-area-network/

[2] Fibre Channel Industry Association. (2018) Fibre Channel Solutions Guide. https://fibrechannel.org/wp-content/uploads/2018/08/FCIA_SolutionsGuide2018_web.pdf

[3] NetApp. (2020) What Is NVMe? https://www.netapp.com/data-storage/nvme/what-is-nvme/

[4] Jens Axboe. (2020) fio - Flexible I/O Tester rev. 3.23. https://fio.readthedocs.io/en/latest/fio_doc.html

[5] Todd Muirhead. (2017, October) Performance of SQL Server 2017 for Linxus VMs on vSphere 6.5. https://blogs.vmware.com/performance/2017/10/performance-sql-server-2017-linux-vms-vsphere-6-5.html

[6] Microsoft. (2020, June) Quickstart: Install SQL Server and Create a Database on RedHat. https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-red-hat?view=sql-server-ver15

[7] Mark Achtemichuk. (2017, March) Virtual Machine vCPU and vNUMA Rightsizing – Guidelines. https://blogs.vmware.com/performance/2017/03/virtual-machine-vcpu-and-vnuma-rightsizing-rules-of-thumb.html

[8] Qasim Ali. (2013, August) Host Power Management in VMware vSphere 5.5. https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/hpm-performance-vsphere55-white-paper.pdf

About the Authors

David Morse is a performance engineer at VMware. He characterizes SQL Server performance within virtual machines on a variety of host and storage solutions. He has 20 years of benchmarking experience between VMware and Dell.

Ravi Krishnamurthy is a performance engineer at VMware. He works on the performance analysis of the storage stack and associated parts of the VMkernel on various solutions. He has 7 years of benchmarking and performance analysis experience and 28 years of total experience.

Acknowledgements

The authors would like to thank NetApp for loaning their storage array to us, NetApp and Broadcom, for their close collaboration in optimizing NVMe-oF performance during the development of vSphere 7.0, and Naveen Krishnamurthy for his comments on this paper. Thanks to Nidhi Naik and Saumyajit Dey for the performance runs and generating and formatting this data.