Technical White Paper

NVMe Transport Performance Comparison

Dell PowerStore: Fibre Channel (FC), NVMe/FC, NVMe/TCP and iSCSI

Abstract

This technical white paper provides the results of NVMe transport performance testing using Dell PowerStore with the same workload configurations to determine performance tradeoffs.

April 2023

Revisions

Date	Description
April 2023 May 2023	Initial release. Simplification release. Pricing and summary updates.

Acknowledgments

The following people made significant contributions to this white paper:

- Gillian Mullins – Dell Technologies Customer Solution Centers - Overall Project Management
- Gerard O'Donovan, Dell Technologies Customer Solution Centers - All testing and re-testing
- Keith Snell – Data validation, analysis, and visualization
- Erik Smith – NVMe/TCP Ecosystem Architect
- Jonathan Tang – Configuration validation
- Eldad Zinger – PowerStore architecture

Dell Technologies Customer Solution Centers help customers explore and validate the Dell Technologies portfolio of hardware, software and solutions. Each of our global Customer Solution Centers is staffed by deep subject matter experts, working in state-of-the-art labs equipped with the latest technologies and showcases.

Services provided by Customer Solution Centers include technical briefings, architectural design workshops and proof-of-concepts that allow customers to validate their chosen solution.

Dell customers may engage with their account team to take advantage of these free services. Learn more at https://www.dell.com/en-us/dt/what-we-do/customer-engagement-programs/customer-solution-centers.htm#tab0=0

Acknowledgments

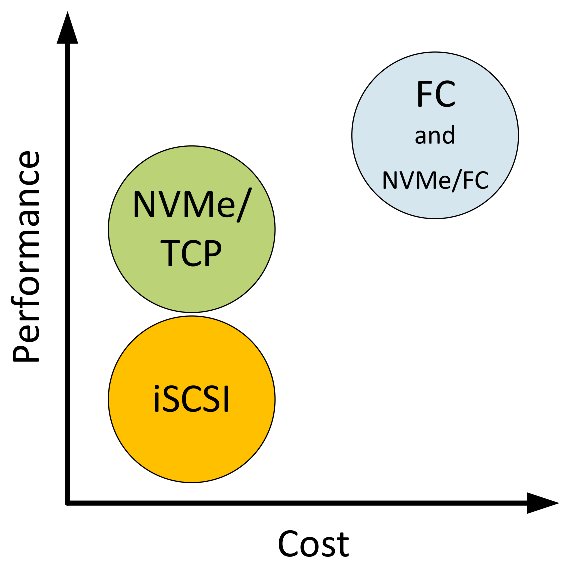
Table of contents

Executive summary

Purpose

The purpose of this white paper is to help Dell PowerStore customers understand the performance and cost tradeoffs when using one Transport Protocol versus another. Throughout the testing we use the same configuration of VMware ESX servers to provide an apple–to-apples comparison of Fibre Channel (traditional FC), NVMe/FC, NVMe/TCP and iSCSI. The configuration used for testing as well as the process we used to generate the load are all provided below to allow others to compare these results to the results they are observing.

Key findings and recommendations



When workloads are running in a VM on VMware ESX that utilize Dell's PowerStore Storage subsystem for external storage capacity, NVMe/TCP provides the lowest total cost per IO and best performance per dollar. Over time, as NVMe/TCP solutions continue to mature and functions like T10-DIF (Data Integrity Field) are offloaded to hardware, you can expect the performance of NVMe/TCP to meet or exceed what is possible with 32- or 64Gigabits per second (Gb/s) Fibre Channel.

When cost is no concern and only raw performance (i.e., IOPS and latency) is considered important, NVMe/FC provides higher IOPS and lower latency than traditional (SCSI-FCP based) Fibre Channel. NVMe/TCP's IOPS and latency (@25 Gbps) are similar to NVMe/FC and FCP (@32GFC) for WRITE commands and within 20% of NVMe/FC and FCP for READ commands. iSCSI provides the lowest IOPS and highest latency.

When other factors (i.e., CPU utilization) are considered most important, the lowest CPU utilization per IO was found when using NVMe/FC. NVMe/TCP was a close second and used less CPU per IO than FCP (Traditional Fibre Channel) in most cases. Finally, iSCSI had the highest overall CPU utilization per IO.

1 Technical White Paper Introduction

1.1 Background

In early 2018 a group of Engineers at Dell were busy trying to reproduce some of the claims being made about NVMe-oF's performance benefits relative to SCSI. We, and many others in the industry, had seen reports that it could improve performance by up to 10x.

After a few quarters of testing, summarized in Figure 1 (below) and described in detail here, we actually did find that a significant performance improvement over existing storage protocols and NVMe/FC was possible, but only when we used NVMe/RoCE (shown as RoCE below) or NVMe/TCP @ 100GbE.

As we compared FC and Ethernet costs, we also realized our customers could take advantage of this performance benefit for less than the cost of FC (see Figure 2).

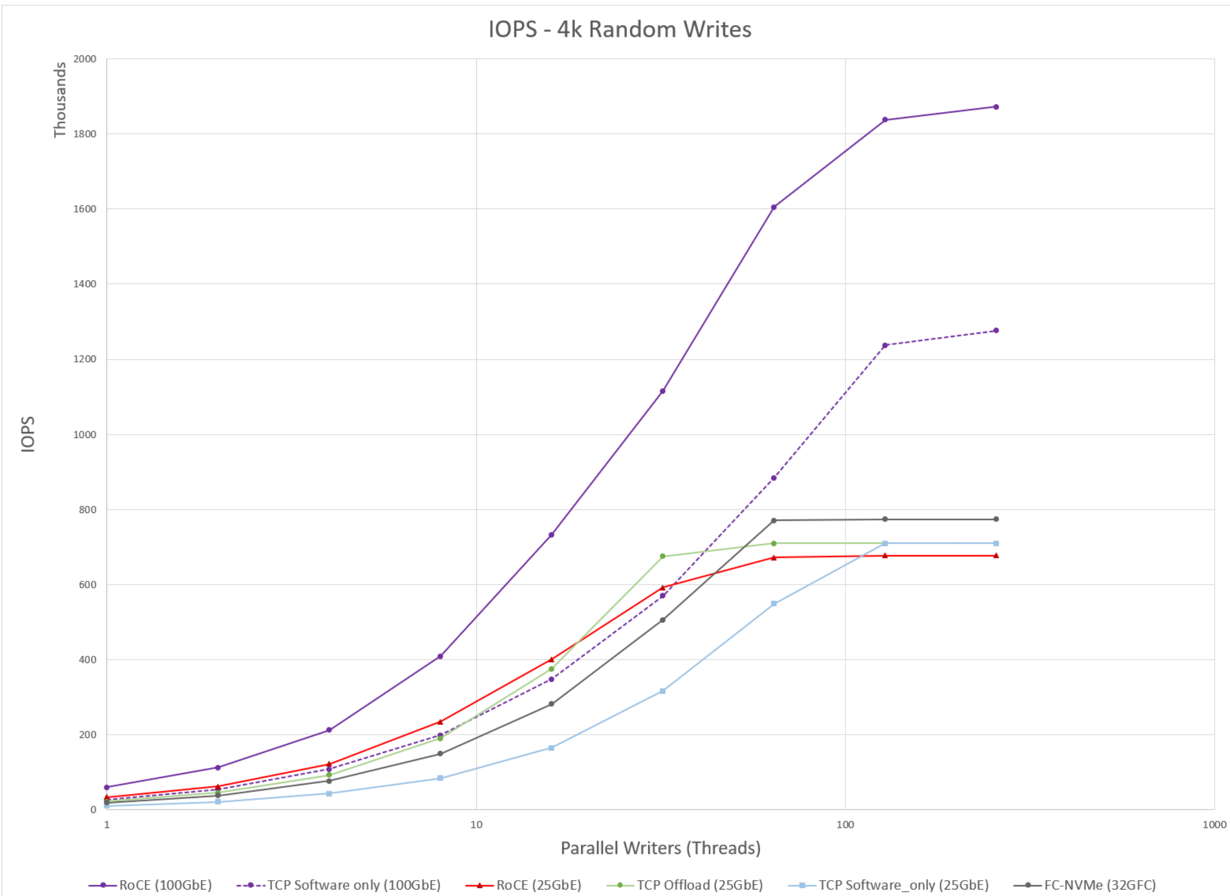


Figure 1 Transport Protocol comparison

Notes:

- *The dots on each line of the graph (above) correspond to the number of threads in use at the time, they are 1, 2, 4, 8, 16, 32, 64, 128, and 256.*

- *The protocol comparisons shown above only compare the transport protocols themselves and should be considered the best performance that is theoretically possible. These comparisons do not consider the impact that storage system architecture will have. Again, see this presentation to understand how they should be interpreted.*
- *We tested 100GbE to determine what was theoretically possible and have included the results here for that purpose. However, the remainder of this paper is focused on comparing 32GFC with 25GbE as 25GbE is the common starting point for NVMe/TCP across our storage products.*
- *Although Dell does use RoCE within some of our storage platform offerings, we chose not to use NVMe/RoCE for general purpose connectivity because of the complexity involved during setup, the extensive tuning required especially when deploying a multi-vendor environment and its susceptibility to Congestion Spreading due to the use of lossless Ethernet.*
- *The chart above shows the benefit of 100GbE and 25GbE both with and without hardware offload. As of today, Dell supports software based NVMe/TCP and is investigating adding support for hardware offloads.*

1.2 Transport Protocol cost comparison

This section compares the cost of the Ethernet and FC transports. For the sake of clarity, please consider the following points as you review the data in figure 2.

- *"FC" used in the comparison represents both traditional FC and NVMe/FC.*
- *"Ethernet" represents both NVMe/TCP and iSCSI.*
- *The CAPEX comparison provided below implicitly treats FC and Ethernet ports as providing equivalent performance. As you will see in the performance data included below, on PowerStore, NVMe/TCP provides nearly equivalent IOPS with Writes and currently up to 20% fewer IOPS with Reads. This needs to be factored into CAPEX considerations that are specific to your environment and workloads.*
- *The CAPEX comparison provides both a "Cost per Gigabit per second" and a "Cost per Port" metric.*
  - *Cost per Gigabit per second is calculated by dividing the Cost by the bandwidth supported by the switch.*
  - *Cost per Port is calculated by dividing the Cost by the number of ports supported by the switch.*
- *Config 1 – This configuration represents the maximum cost difference between Ethernet and FC. this configuration only considers the cost of adapters, transceivers, cabling and switching infrastructure that is required to support Ethernet or FC based IO. This configuration can also be thought of as representing the difference in transport protocol cost when comparing Software Defined Storage (SDS) versus a traditional FC SAN.*
- *Configuration 2 – Represents an end-to-end comparison of a configuration containing 25GbE and 32GbE switches and adapters including PowerStore. This configuration consists of 24 Host interfaces and 8 storage interfaces (4:1 fan-out ratio)*
- *Configuration 3 – Represents the cost of the configuration used to provide the performance metrics in this paper. The cost of each solution is very close because we used 100GbE adapters for the test (Note the adapters were running at 25GbE). 100GbE adapters were used because we had them available and were not optimizing the configuration for cost. If we had used 25GbE adapters, the comparison would be close to Config 2.*
- *After reviewing the performance and CPU utilization data, NVMe/TCP at 25GbE offers relatively lower performance that is offset by the significant costs savings using Ethernet infrastructure*

After taking all the above into account, they key takeaway from our analysis was:

For network infrastructure deployment, Ethernet cost is up to 81% less than FC in some configurations

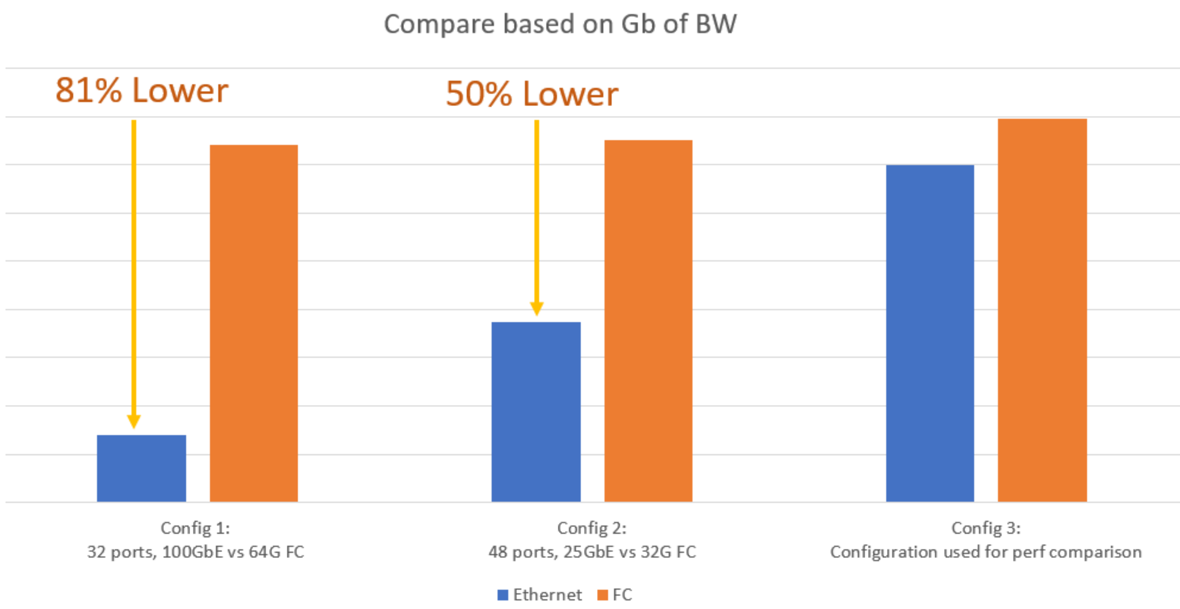


Figure 2 Cost comparison based on Gb of Bandwidth FC and Ethernet

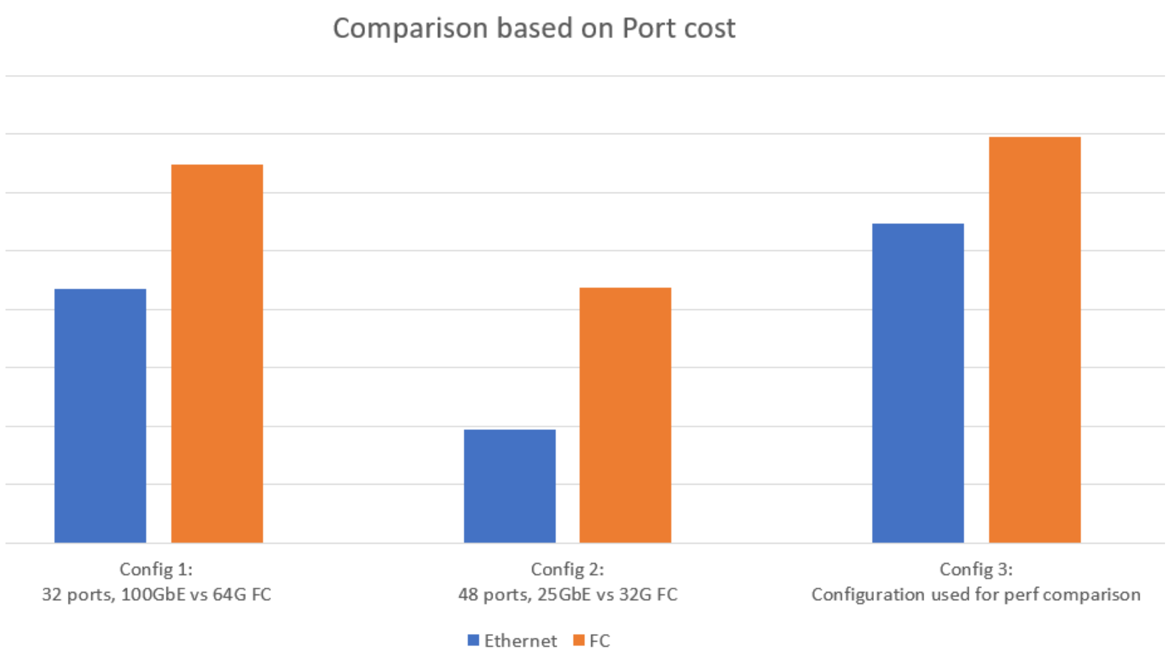


Figure 2a Cost comparison based on port cost of FC and Ethernet

Perhaps we shouldn't have been surprised by the test results or the cost comparison, but the results were so dramatic, we wondered "Why haven't customers already started to move off traditional transport protocols and onto NVMe/RoCE or NVMe/TCP"?

From our experience speaking with SAN administrators, we know they are concerned about stability, security and manageability. This makes complete sense because in some operating environments a loss of connectivity to storage can have severe business consequences.

As a result, we knew if these customers ever decided to move off FC and onto Ethernet, they would only do so if some basic SAN features were made available for use with Ethernet. These features include:

- A user experience similar to FC
- Automated Discovery of fabric services and storage capacity
- Access control (similar to Hard Zoning and LUN Masking)
- Support for Authentication and Encryption
- Performance that is comparable to FC
- Wide industry/ecosystem support

If all these features could be provided, a lower cost solution might cause them to think about moving to a new, Ethernet based, transport for net new or replacement infrastructure deployments. With these requirements in mind, Dell decided to be proactive and engineer an IP-Based SAN solution ready for our customers to adopt. Since starting this initiative, we've addressed many of these concerns by:

1. Working with our competitors in the industry to create a standardized Centralized Discovery Controller (CDC). Note: Dell was first to market with our CDC implementation known as SmartFabric Storage Software (SFSS).
2. To address security concerns Dell has led the security standardization efforts. First with TP8006 and TP8011 that added support for Authentication and Secure channels respectively. More recently, with TP8019, it standardized a new Centralized Authentication Verification Entity (AVE) concept that will allow end-users to easily manage security at a scale that has never been experienced by SAN administrators before.
3. We have also started the process of enhancing NVMe/TCP performance even though it's much better than iSCSI today. You can expect to see many performance related enhancements coming soon.

It's also worth mentioning that Ethernet's rich vendor ecosystem is invested in by the Hyperscaler community. As a result, Ethernet/IP based connectivity will continue to evolve to support new use cases, while simultaneously decreasing in price on a dollars per Gb basis.

2 Performance Testing

2.1 Overview

Information about the testing performed for this white paper is described in this section. This information includes:

- The configurations tested
- Simplified Test results, and
- Vdbench (load generator) settings

All are being provided to allow Dell PowerStore users to understand the performance tradeoffs when using one Transport Protocol versus another.

Throughout the testing we use the same configuration of VMware ESX servers to provide an apples–to-apples comparison of traditional Fibre Channel (FCP), NVMe over Fibre Channel (NVMe/FC), NVMe over TCP (NVMe/TCP) and iSCSI. The configurations used during our testing as well as the scripts used to generate the load are all provided below to allow our customers to use them as a baseline for the sake of this comparison.

Unless otherwise specified, the IOPS, Latency and CPU measurements provided in this paper are all based on default operating parameters for both ESX and PowerStore. Accordingly, these test results do not represent the absolute best performance possible with either ESX or PowerStore, instead, they represent what most users can expect from each of the protocols involved in the study. Because of this, we left the ESX Multipathing policy set to Round Robin even though we discovered that setting it to IOPS led to better performance under certain conditions (e.g., fabric congestion).

This white paper contains simplified performance results, for complete test results, please reach out to your Dell Account representative.

2.2 Test Configurations

Each of the following test configurations consists of:

- (4) ESX Servers (running ESX 7.0u3), with each server having:
  - (2) VMs with VDBench running within each VM (8 VMs total)
  - (16) (100GB) PowerStore volumes assigned to each Server (8 per VM)
  - (2) Network adaptors (either FC or Ethernet)

- (1) PowerStore 9000 (version 3.0) consisting of
  - 64 (100GB volumes) for FC and iSCSI tests
  - 64 (100GB volumes) for NVMe/FC and NVMe/TCP testing
  - 16 Front end interfaces (either FC or Ethernet)
- A network consisting of either:
  - Two Broadcom G720 64Gb-capable FC switches, running with 32Gb/s SFPs, or
  - Two Dell Networking S5248F Ethernet/IP switches

Each of the interfaces and storage volumes have been color coded to help illustrate how each Host interface is connected to the PowerStore and how each volume is presented to the hosts. While the exact connectivity details are not critical, it is important to note that since the purpose of this testing was to allow customers to understand how each transport protocol can be expected to perform, we did not include any test scenarios that involved impairments of any kind. This not only includes dropping packets or injecting bit errors, but also configuring the system to ensure that each host could have 100% of the bandwidth available from each storage interface.

To accomplish this:

- Each host interface is configured to access a dedicated storage interface to create a 1:1 non-oversubscribed configuration,
- The host interface and the storage interface it accesses are connected to the same physical switch as a part of the same VLAN, and
- We have explicitly configured node affinity for each PowerStore storage volume to ensure that it will only be accessed via one interface on one Node.

2.2.1 Fibre Channel

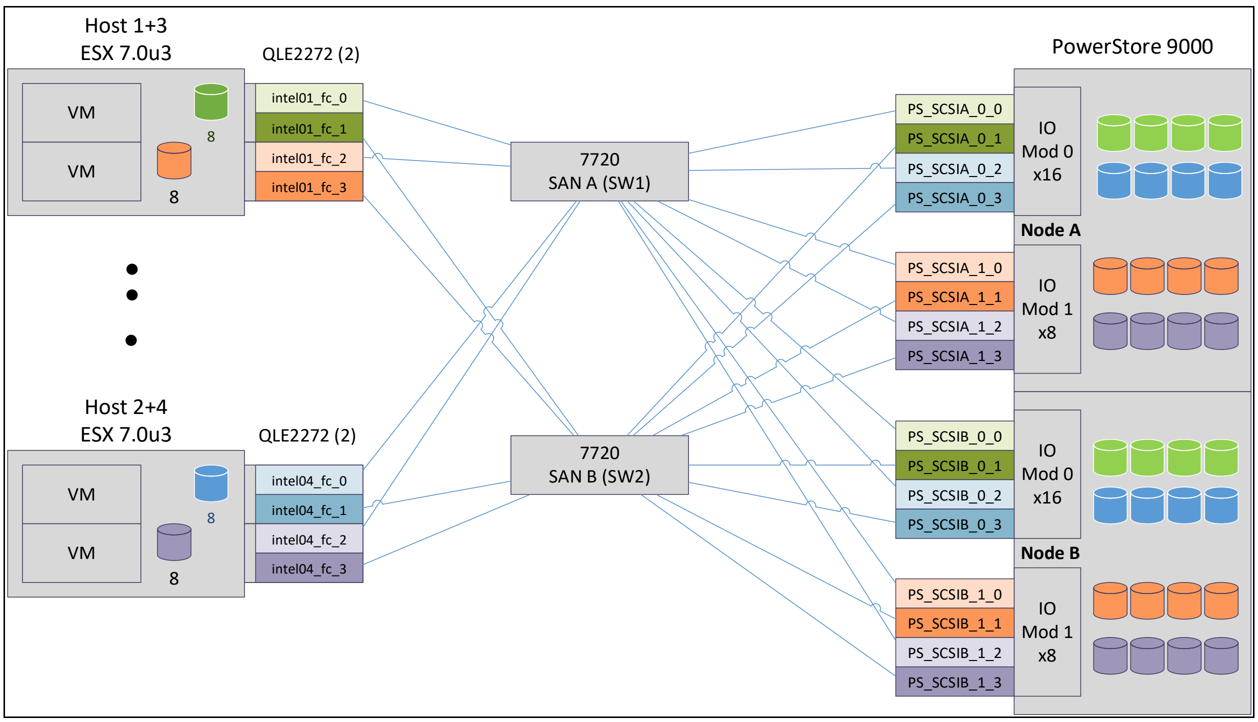


Figure 3 Fibre Channel configuration used for testing

2.2.2 Ethernet/IP

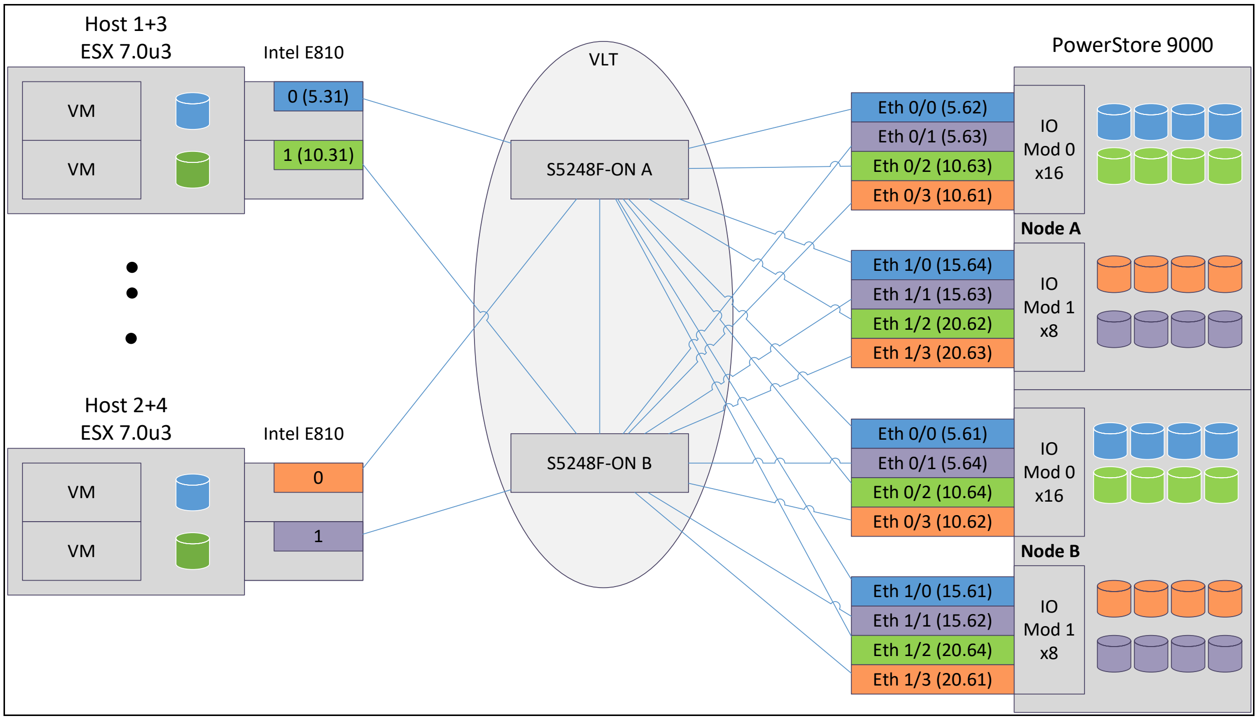


Figure 4 Ethernet/IP configuration used for testing

3 Performance Comparison Results

The following section provides a comparison of four storage transport protocols (FC, NVMe/FC, NVMe/TCP and iSCSI) that can be used by a Host to access external (array based) storage capacity. Each protocol is compared using three measurements, IOPS, Latency and CPU Utilization.

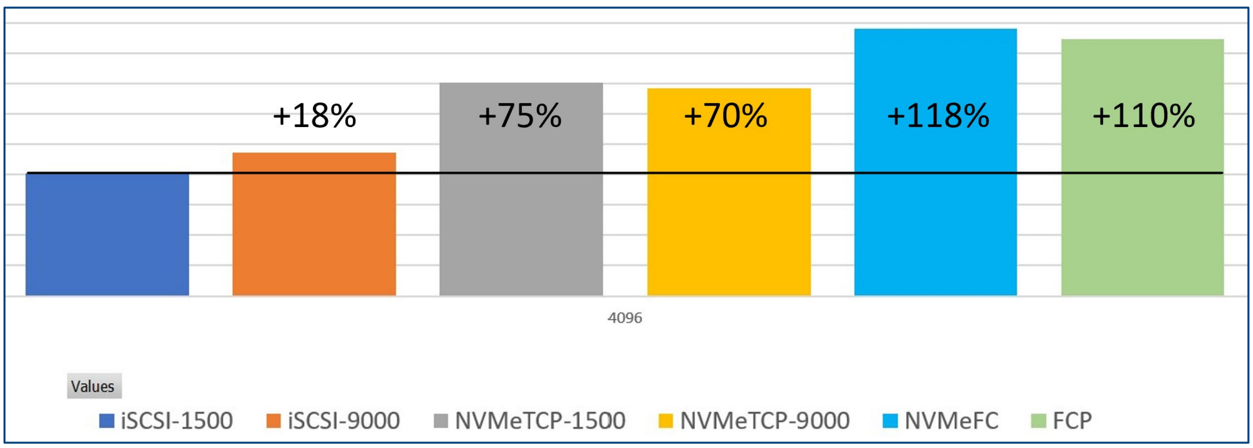
This white paper contains simplified performance results, for complete test results, please reach out to your Dell Account representative. All performance results provided in this version of the white paper utilized a block size of 4k.

3.1 IOPS Overview

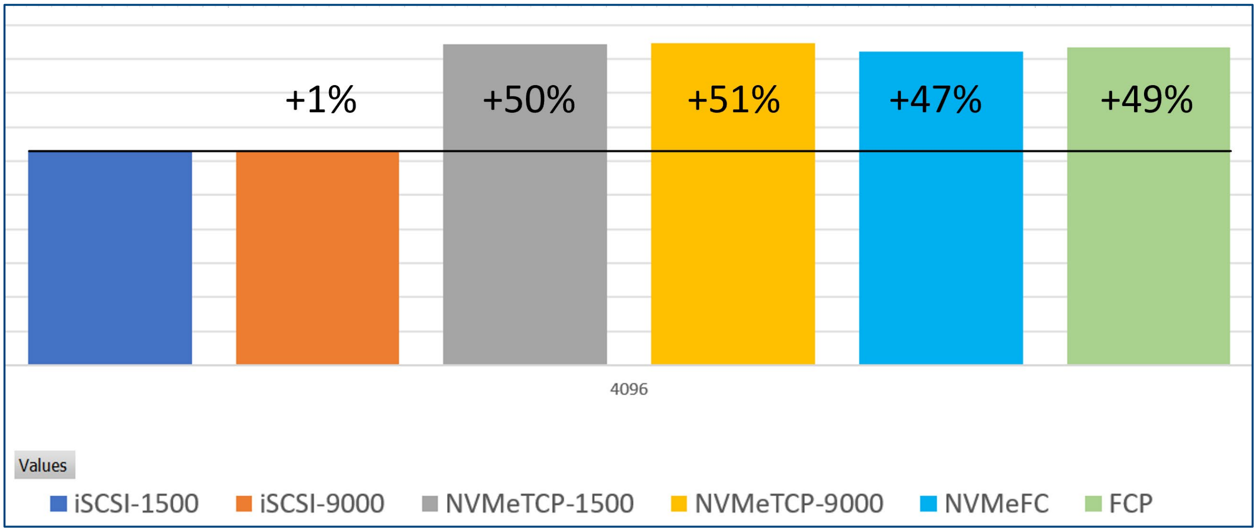
The results for IOPS should be interpreted as follows:

- Starting from the left of each figure, each bar corresponds to a particular transport protocol that was tested:
    - **iSCSI-1500**: iSCSI with the MTU set to 1500 bytes
    - **iSCSI-9000**: iSCSI with the MTU set to 9000 bytes
    - **NVMeTCP-1500**: NVMe/TCP with the MTU set to 1500 bytes
    - **NVMeTCP-9000**: NVMe/TCP with the MTU set to 9000 bytes
    - **NVMeFC**: NVMe/FC (NVME over FC)
    - **FCP**: SCSI-FCP (Traditional- SCSI based FC)
- The percentage shown above each bar represents the % difference from our chosen baseline (iSCSI-1500). A positive percentage indicates that the transport supported more IOPS than iSCSI-1500. A negative percentage indicates that the transport supported less IOPS that iSCSI-1500. **Higher is better.**

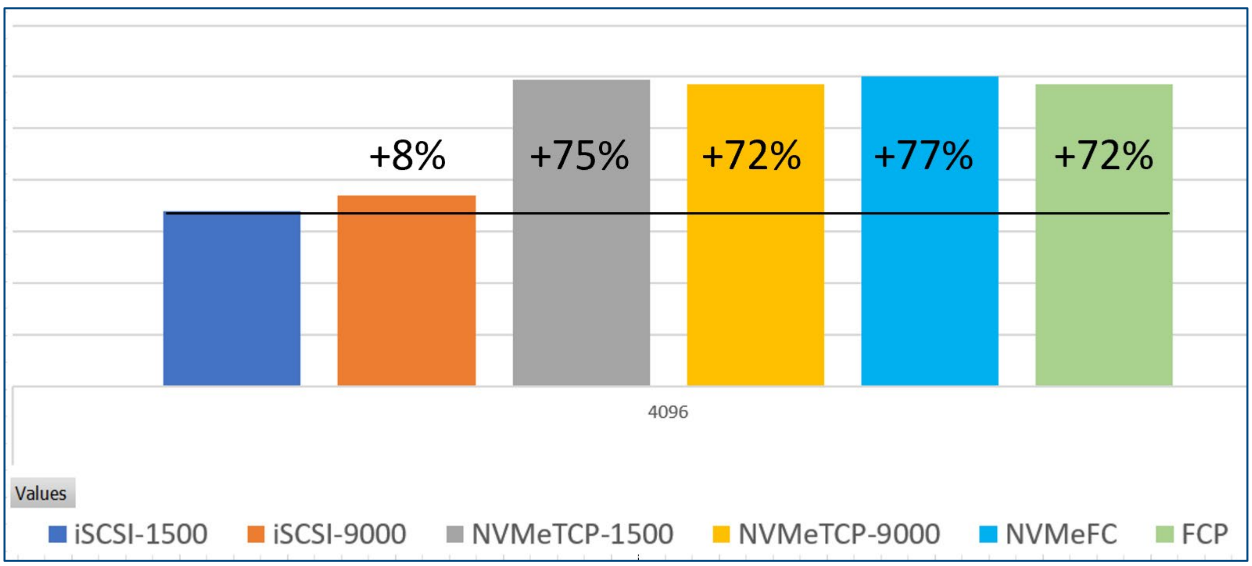
3.1.1 IOPS – 4K - 100% READ



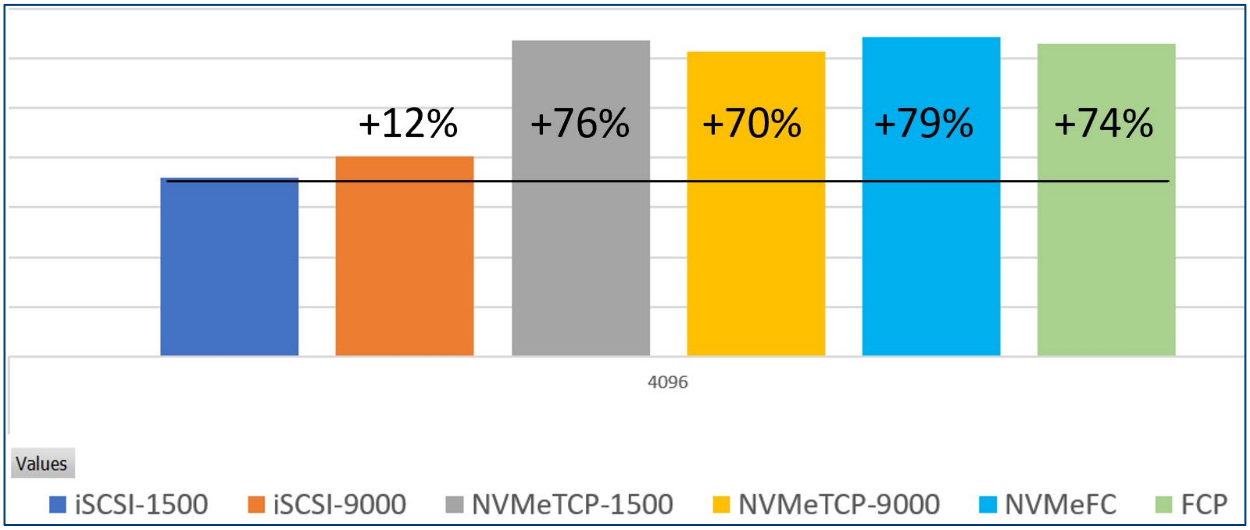
3.1.2 IOPS – 4K - 100% WRITE



3.1.3 IOPS – 4K - 50% READ / 50% WRITE



3.1.4 IOPS – 4K - 70% READ / 30% WRITE

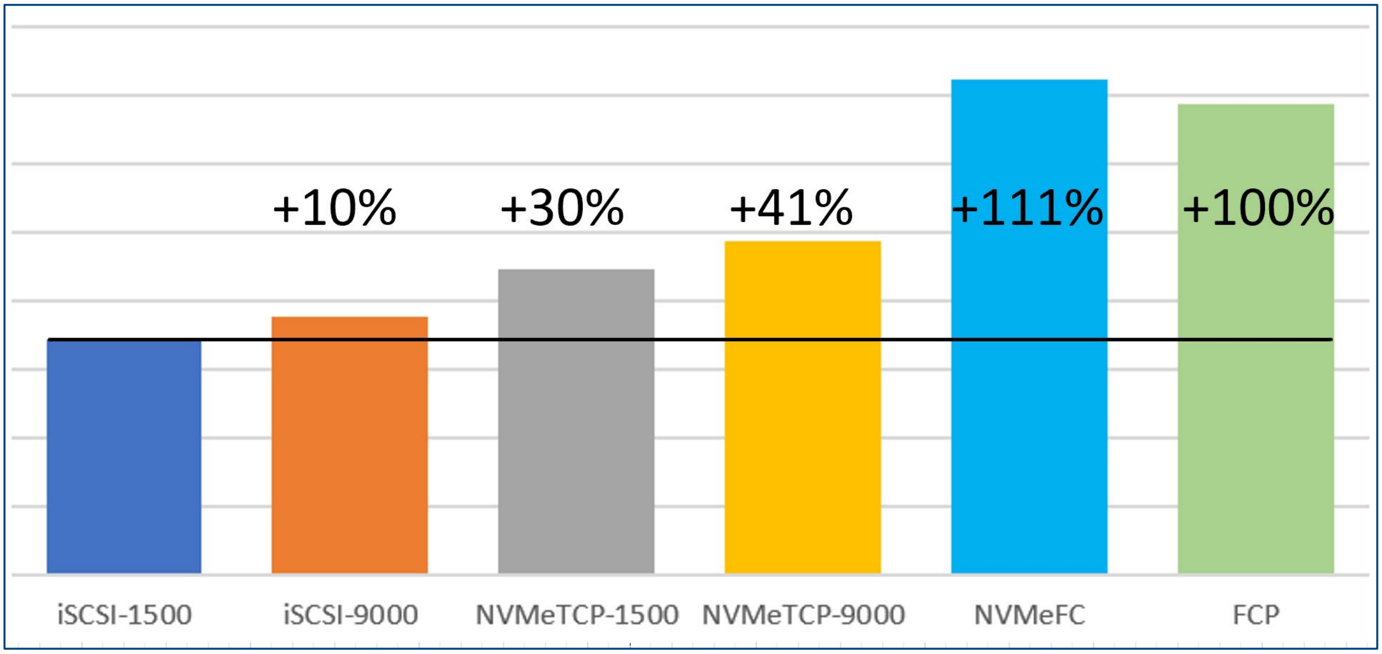


3.2 I/O before saturation Overview

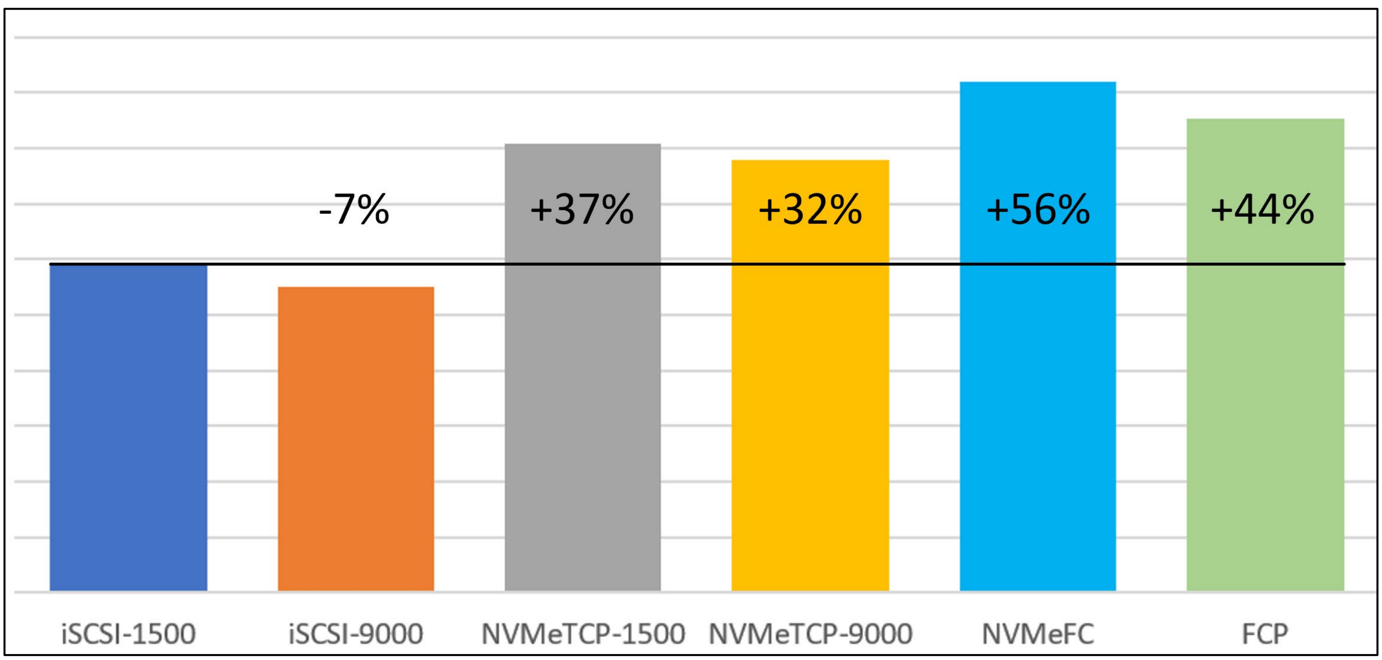
The results for I/O before saturation may be interpreted as follows:

- Starting from the left of each figure, each bar corresponds to a particular transport protocol that was tested:
    - **iSCSI-1500**: iSCSI with the MTU set to 1500 bytes
    - **iSCSI-9000**: iSCSI with the MTU set to 9000 bytes
    - **NVMeTCP-1500**: NVMe/TCP with the MTU set to 1500 bytes
    - **NVMeTCP-9000**: NVMe/TCP with the MTU set to 9000 bytes
    - **NVMeFC**: NVMe/FC (NVME over FC)
    - **FCP**: SCSI-FCP (Traditional- SCSI based FC)
- The percentage shown above each bar represents the % difference from our chosen baseline (iSCSI-1500). A positive percentage indicates that the transport is capable of supporting more IOPS before saturation than iSCSI-1500. A negative percentage indicates that the transport supported less IOPS before saturation than iSCSI-1500.
- **Higher is better**.
- For the purposes of this white paper, assume that saturation occurs when the graph of a curve hits a knee, and the y-axis values increase non-linearly.

3.2.1 I/O before saturation – 4K 100% READ



3.2.2 I/O before saturation 4K 100% WRITE

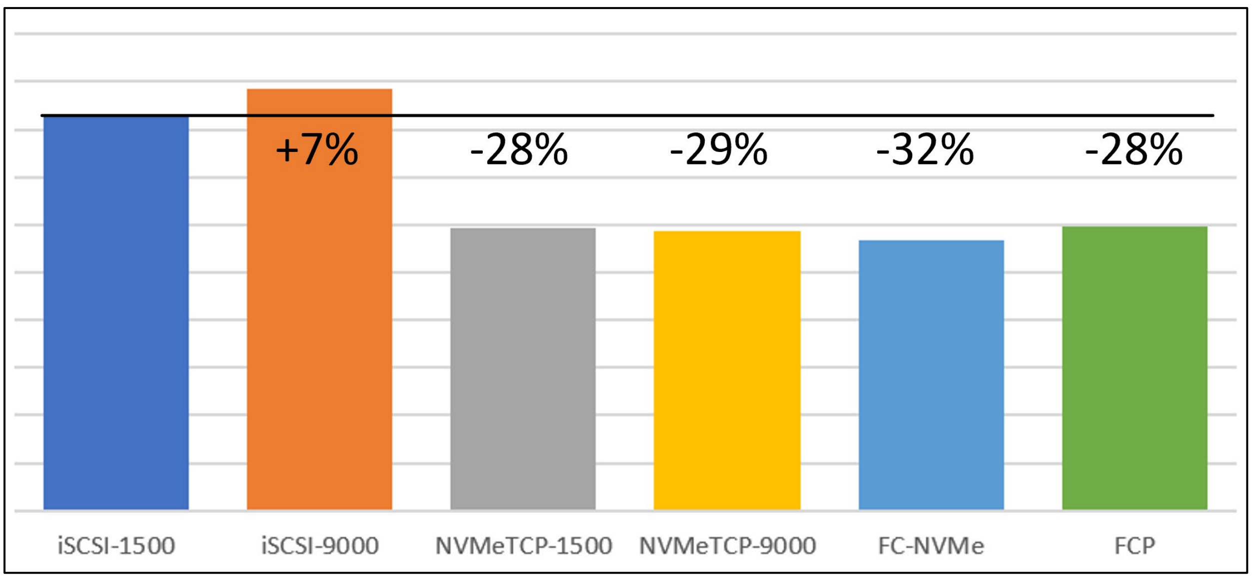


3.3 CPU Utilization Overview

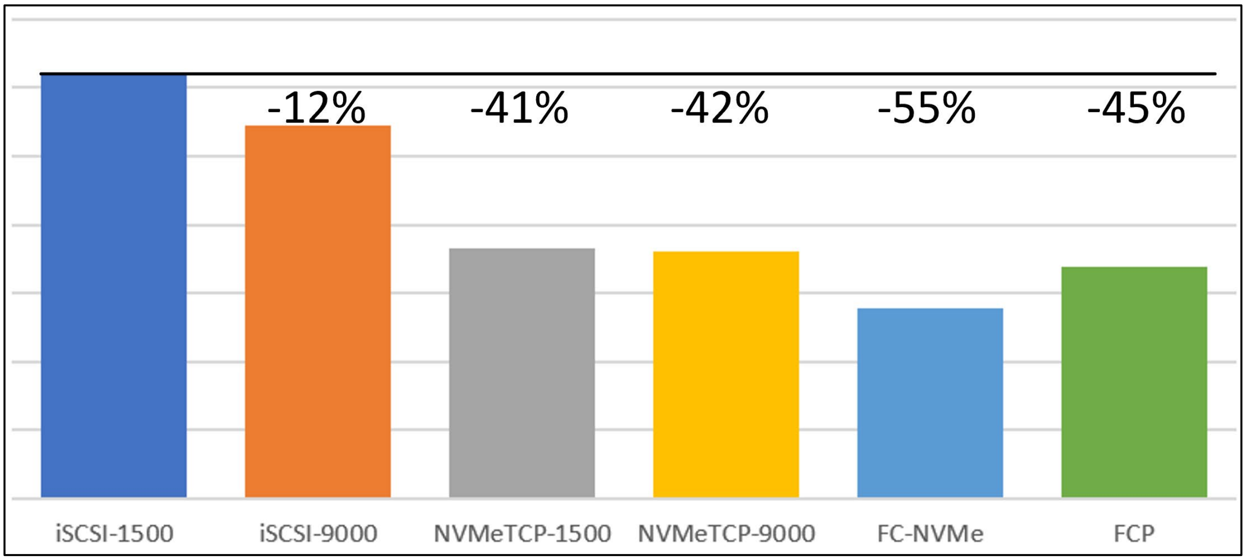
The results for CPU utilization may be interpreted as follows:

- Starting from the left of each figure, each bar corresponds to a particular transport protocol that was tested:
  - **iSCSI-1500**: iSCSI with the MTU set to 1500 bytes
  - **iSCSI-9000**: iSCSI with the MTU set to 9000 bytes
  - **NVMeTCP-1500**: NVMe/TCP with the MTU set to 1500 bytes
  - **NVMeTCP-9000**: NVMe/TCP with the MTU set to 9000 bytes
  - **NVMeFC**: NVMe/FC (NVME over FC)
  - **FCP**: SCSI-FCP (Traditional- SCSI based FC)
- The percentage shown above each bar represents the % difference from our chosen baseline (iSCSI-1500). A positive percentage indicates that the transport uses more CPU per I/O. A negative percentage indicates that the transport uses less CPU per IO than iSCSI-1500.
- The percentage of CPU utilization per IO was calculated by taking the average CPU utilization during the test and dividing this by the Average IOPS observed during the same period of time.
- Lower is better.

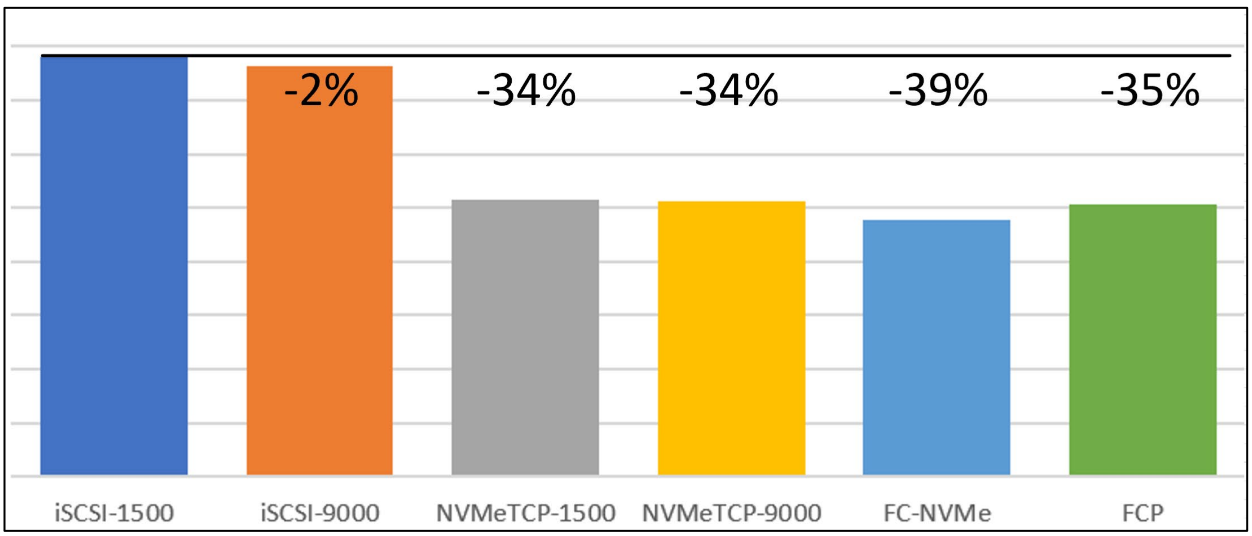
3.3.1 CPU Utilization – 4K - 100% Write



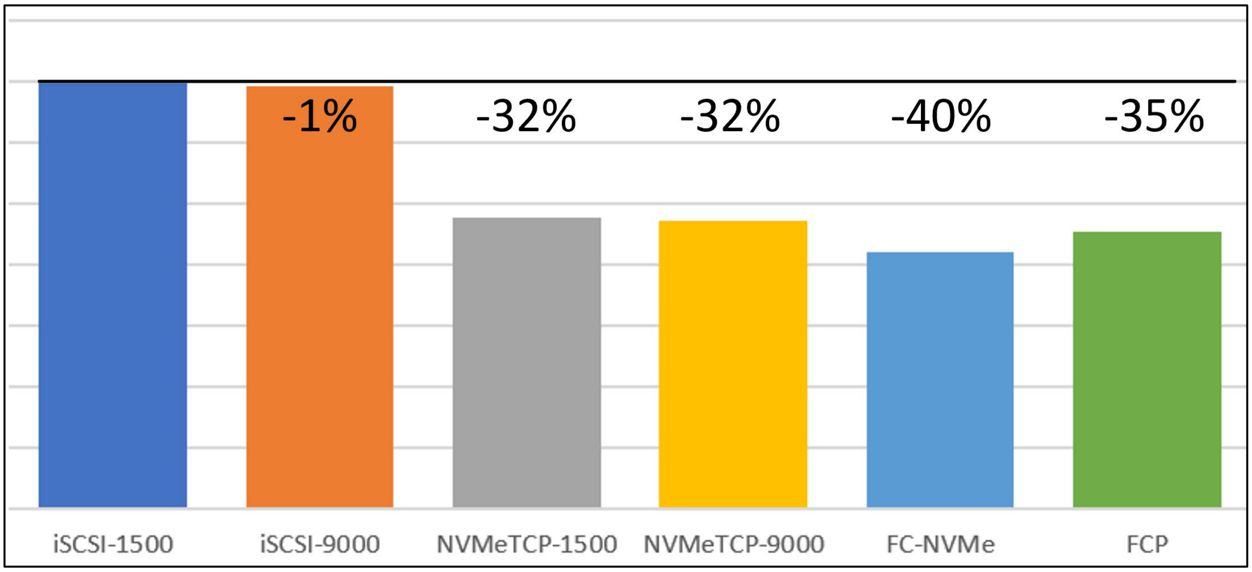
3.3.2 CPU Utilization – 4K - 100% Read



3.3.3 CPU Utilization – 4K - 50% READ / 50% WRITE



3.3.4 CPU Utilization – 4K - 70% READ / 30% WRITE



4 NVMe/TCP performance improvements - Futures

Although NVMe/TCP @25GbE is nearly equivalent to NVMe/FC @32GFC, there's plenty of room for improvement especially when considering the impact that support for higher speeds (i.e., 100GbE) and Secure Channels (i.e., TLS 1.3) will have on the hosts ability to maintain throughput due to the load on the system CPU. To offset this load, Dell and many other companies in the industry are investigating the use of Data Processing Units (DPUs) to allow for different tiers of performance (e.g., cost optimized, performance optimized, etc). Expect to see much more on this topic in the future.

5 Conclusion

We believe NVMe/TCP is a practical alternative to iSCSI. With NVMe/TCP's ability to provide higher IOPS at a lower latency while consuming less CPU than iSCSI, NVMe/TCP can provide an immediate benefit.

We also believe that with the addition of Fibre Channel like services (e.g., SmartFabric Storage Software) NVMe/TCP is becoming a practical alternative to Fibre Channel, especially with NVMe/TCP's ability to support Edge and Software Defined Storage use cases. In addition, for customers who are under cost constraints and are being challenged to move off of FC and onto Ethernet, the combination of NVMe/TCP and Dell's SmartFabric Storage Software (SFSS) can provide a starting point without having to re-learn a completely new set of SAN management paradigms.

Dell has also invested extensively in the NVMe/TCP ecosystem to ensure that security and end-to-end visibility of storage traffic and flows will be available for our customers to use.

.

5.1 NVMe Protocol Decision Table

When to consider NVMe/Fibre Channel	When to consider NVMe/TCP
Existing FC environment needs to add capacity (ports)	Existing iSCSI environment needs to add capacity (ports)
Highest number of IOPS are required	Software Defined Storage support is required
Lowest latency is required	Edge/distributed systems at scale is required
Lowest CPU utilization per IO is required	Cloud operating model is required
CAPEX is not the most important factor	CAPEX is a very important factor

Resources

Introduction to NVMe over TCP and SmartFabric Storage Software Video

The NVMe/TCP Dating App Blog

6 Testing process

6.1 Basic configuration details

Since the purpose of this testing was to allow customers to understand how each transport protocol can be expected to perform, we did not include any test scenarios that involved impairments of any kind. This not only includes dropping packets or injecting bit errors, but also configuring the system to ensure that each host could have 100% of the bandwidth available from each storage interface. To accomplish this:

- Each host interface is configured to access a dedicated storage interface (1:1 non-oversubscribed configuration),
- The host interface and the storage interface is accesses are connected to the same physical switch and are located in the same VLAN, and
- We have explicitly configured node affinity for each PowerStore storage volume to ensure that it will only be accessed via one interface on one Node.

6.2 Test steps

All the performance metrics provided in this paper were captured using vdbench. Since we wanted to provide customers with expectations about the overall impact on their systems when using one transport protocol versus another, we decided to run vdbench client instances in each of our 8 VMs and then kick off the testing from a single centralized location. This allowed us to collect stats from all clients at the same time and correlate the results.

Each VM used for testing was running Ubuntu and needed to have /etc/hosts updated to allow them to reach the centralized location that was kicking off the script. Each VM had 8-100G datastores mapped to it.

Each transport protocol was evaluated using the same set of parameters:

- Read/Write ratios: 0, 50, 70, 100
- Block sizes: 4K, 8k, 16K, 32K
- Number of threads: 1, 2, 4, 8, 16, 32

Before each test was performed a prefill script was used to ensure background processing tasks on the array (e.g., garbage collection) wouldn't skew test results. The vdbench configuration files for pre-fill and testing are provided below. Please note you will need to modify these files in order for them to work in your environment.

6.2.1 Pre-fill configuration

```
compratio=2,
dedupratio=2,
dedupunit=4k,
dedupsets=5%
messagescan=no

#* ----- Hd Definition -----
hd=default,shell=vdbench,user=root,jvms=8
hd=ubuntu-vm01,system=<IP Address vm01>
hd=ubuntu-vm02,system=<IP Address vm02>
hd=ubuntu-vm03,system=<IP Address vm03>
hd=ubuntu-vm04,system=<IP Address vm04>
hd=ubuntu-vm05,system=<IP Address vm05>
```

```
hd=ubuntu-vm06,system=<IP Address vm06>
hd=ubuntu-vm07,system=<IP Address vm07>
hd=ubuntu-vm08,system=<IP Address vm08>

sd=vol33,hd=ubuntu-vm01,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol34,hd=ubuntu-vm01,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol35,hd=ubuntu-vm01,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol36,hd=ubuntu-vm01,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol37,hd=ubuntu-vm01,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol38,hd=ubuntu-vm01,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol39,hd=ubuntu-vm01,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol40,hd=ubuntu-vm01,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol41,hd=ubuntu-vm02,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol42,hd=ubuntu-vm02,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol43,hd=ubuntu-vm02,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol44,hd=ubuntu-vm02,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol45,hd=ubuntu-vm02,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol46,hd=ubuntu-vm02,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol47,hd=ubuntu-vm02,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol48,hd=ubuntu-vm02,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol49,hd=ubuntu-vm03,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol50,hd=ubuntu-vm03,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol51,hd=ubuntu-vm03,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol52,hd=ubuntu-vm03,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol53,hd=ubuntu-vm03,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol54,hd=ubuntu-vm03,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol55,hd=ubuntu-vm03,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol56,hd=ubuntu-vm03,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol57,hd=ubuntu-vm04,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol58,hd=ubuntu-vm04,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol59,hd=ubuntu-vm04,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol60,hd=ubuntu-vm04,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol61,hd=ubuntu-vm04,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol62,hd=ubuntu-vm04,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol63,hd=ubuntu-vm04,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol64,hd=ubuntu-vm04,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol65,hd=ubuntu-vm05,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol66,hd=ubuntu-vm05,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol67,hd=ubuntu-vm05,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol68,hd=ubuntu-vm05,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol69,hd=ubuntu-vm05,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol70,hd=ubuntu-vm05,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol71,hd=ubuntu-vm05,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol72,hd=ubuntu-vm05,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol73,hd=ubuntu-vm06,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol74,hd=ubuntu-vm06,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol75,hd=ubuntu-vm06,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol76,hd=ubuntu-vm06,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol77,hd=ubuntu-vm06,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol78,hd=ubuntu-vm06,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol79,hd=ubuntu-vm06,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol80,hd=ubuntu-vm06,lun=/dev/sdi,size=100G,openflags=o_direct
```

```
sd=vol81,hd=ubuntu-vm07,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol82,hd=ubuntu-vm07,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol83,hd=ubuntu-vm07,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol84,hd=ubuntu-vm07,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol85,hd=ubuntu-vm07,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol86,hd=ubuntu-vm07,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol87,hd=ubuntu-vm07,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol88,hd=ubuntu-vm07,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol89,hd=ubuntu-vm08,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol90,hd=ubuntu-vm08,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol91,hd=ubuntu-vm08,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol92,hd=ubuntu-vm08,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol93,hd=ubuntu-vm08,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol94,hd=ubuntu-vm08,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol95,hd=ubuntu-vm08,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol96,hd=ubuntu-vm08,lun=/dev/sdi,size=100G,openflags=o_direct


#* ----- Workload Definition : Jobs -----
wd=wVolFilling,sd=vol*,rdpct=0,xfersize=256k,seekpct=eof

#* ----- Run Definition : execution ----
rd=rFilling,wd=wVolFilling,elapsed=50000,interval=10,iorate=max,threads=8
```

6.2.2 vdbench testing parameters

```
compratio=2,
dedupratio=2,
dedupunit=4k,
dedupsets=5%
messagescan=no

#* ----- Hd Definition -----
hd=default,shell=vdbench,user=root,jvms=8
hd=ubuntu-vm01,system=<IP Address vm01>
hd=ubuntu-vm02,system=<IP Address vm02>
hd=ubuntu-vm03,system=<IP Address vm03>
hd=ubuntu-vm04,system=<IP Address vm04>
hd=ubuntu-vm05,system=<IP Address vm05>
hd=ubuntu-vm06,system=<IP Address vm06>
hd=ubuntu-vm07,system=<IP Address vm07>
hd=ubuntu-vm08,system=<IP Address vm08>

sd=vol33,hd=ubuntu-vm01,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol34,hd=ubuntu-vm01,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol35,hd=ubuntu-vm01,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol36,hd=ubuntu-vm01,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol37,hd=ubuntu-vm01,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol38,hd=ubuntu-vm01,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol39,hd=ubuntu-vm01,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol40,hd=ubuntu-vm01,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol41,hd=ubuntu-vm02,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol42,hd=ubuntu-vm02,lun=/dev/sdc,size=100G,openflags=o_direct
```

```
sd=vol43,hd=ubuntu-vm02,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol44,hd=ubuntu-vm02,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol45,hd=ubuntu-vm02,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol46,hd=ubuntu-vm02,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol47,hd=ubuntu-vm02,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol48,hd=ubuntu-vm02,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol49,hd=ubuntu-vm03,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol50,hd=ubuntu-vm03,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol51,hd=ubuntu-vm03,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol52,hd=ubuntu-vm03,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol53,hd=ubuntu-vm03,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol54,hd=ubuntu-vm03,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol55,hd=ubuntu-vm03,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol56,hd=ubuntu-vm03,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol57,hd=ubuntu-vm04,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol58,hd=ubuntu-vm04,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol59,hd=ubuntu-vm04,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol60,hd=ubuntu-vm04,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol61,hd=ubuntu-vm04,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol62,hd=ubuntu-vm04,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol63,hd=ubuntu-vm04,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol64,hd=ubuntu-vm04,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol65,hd=ubuntu-vm05,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol66,hd=ubuntu-vm05,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol67,hd=ubuntu-vm05,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol68,hd=ubuntu-vm05,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol69,hd=ubuntu-vm05,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol70,hd=ubuntu-vm05,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol71,hd=ubuntu-vm05,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol72,hd=ubuntu-vm05,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol73,hd=ubuntu-vm06,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol74,hd=ubuntu-vm06,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol75,hd=ubuntu-vm06,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol76,hd=ubuntu-vm06,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol77,hd=ubuntu-vm06,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol78,hd=ubuntu-vm06,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol79,hd=ubuntu-vm06,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol80,hd=ubuntu-vm06,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol81,hd=ubuntu-vm07,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol82,hd=ubuntu-vm07,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol83,hd=ubuntu-vm07,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol84,hd=ubuntu-vm07,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol85,hd=ubuntu-vm07,lun=/dev/sdf,size=100G,openflags=o_direct
sd=vol86,hd=ubuntu-vm07,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol87,hd=ubuntu-vm07,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol88,hd=ubuntu-vm07,lun=/dev/sdi,size=100G,openflags=o_direct

sd=vol89,hd=ubuntu-vm08,lun=/dev/sdb,size=100G,openflags=o_direct
sd=vol90,hd=ubuntu-vm08,lun=/dev/sdc,size=100G,openflags=o_direct
sd=vol91,hd=ubuntu-vm08,lun=/dev/sdd,size=100G,openflags=o_direct
sd=vol92,hd=ubuntu-vm08,lun=/dev/sde,size=100G,openflags=o_direct
sd=vol93,hd=ubuntu-vm08,lun=/dev/sdf,size=100G,openflags=o_direct
```

```
sd=vol94,hd=ubuntu-vm08,lun=/dev/sdg,size=100G,openflags=o_direct
sd=vol95,hd=ubuntu-vm08,lun=/dev/sdh,size=100G,openflags=o_direct
sd=vol96,hd=ubuntu-vm08,lun=/dev/sdi,size=100G,openflags=o_direct


#* ----- Workload Definition : Jobs -----
wd=wd_ran,seekpct=rand

#* ----- Run Definition : execution -----
rd=rd_ran,wd=wd_ran,sd=vol*,iorate=max,warmup=20,pause=20,elapsed=60,interval=5,xfersize=(4k, 8K,
16K, 32K),rdpct=(0,50,70,100),threads=(1,2,4,8,16,32)
```