BROCADE
A Broadcom Company

EMULEX®
Fibre Channel HBAs

PURESTORAGE®

Application Testing of NVMe over Fibre Channel with Broadcom and Pure Storage: White Paper

Overview

Every organization that utilizes business-critical applications faces the never-ending challenge of increased demands on their IT infrastructure while resources and budgets continue to shrink. Technology is always evolving and innovating to enable customers in meeting this challenge. The question then becomes: *Which technologies will provide increased productivity without compromising or impacting the bottom line?* A second and probably more important question is: *Can these new technologies be deployed without disrupting my business or processes that are needed to service my customers?* In this paper, you will see how dramatic performance increases can be realized non-disruptively and in many cases without any new investments.

NVM Express (NVMe) is a high performance, low latency, storage protocol designed from the ground up for Solid State Disks (SSDs). Since its introduction in 2011, NVMe has quickly overtaken the Small Computer System Interface (SCSI) command protocol as the protocol of choice for direct-attached storage. Building on that success, in 2014 the NVM Express Organization began work to scale NVMe to meet the needs of the enterprise data center. This work led to the NVMe over Fabrics (NVMe-oF) specification, initially published in 2016. NVMe-oF is an extension of the NVMe protocol to Ethernet and Fibre Channel networks—delivering faster and more efficient connectivity between storage and servers.

In the data center, Fibre Channel (FC) has long been the storage networking transport of choice for mission-critical storage applications. FC was designed and purpose-built for storage and has proven to be reliable and stable, while providing ultra-low latency yielding deterministic application performance. Three generations of Brocade FC switches and directors have supported NVMe traffic in production environments. Each generation of FC has built on the previous, adding additional telemetry, reporting, and manageability unique to NVMe traffic. FC allows for SCSI I/O operations and NVMe I/O operations to be used simultaneously in the same SAN fabric. This enables a data center to add new NVMe workloads and to migrate critical SCSI workloads to NVMe without disrupting the remaining SCSI workloads. Simply put, customers can non-disruptively realize the application performance benefits of NVMe over FC (FC-NVMe) with no hardware rip-and-replace required.

All Brocade® FC Storage Area Network (SAN) switches running FOS 8.1 or later support FC-NVMe and SCSI over FC (FC-SCSI) concurrently on all ports at all supported speeds. However, the implementation of FC-NVMe requires more than just the Brocade switches being able to transport FC-NVMe. It requires support by the storage array, the Host Bus Adapter (HBA), and the server operating system. The good news is this capability is available today:

- Emulex® supports FC-NVMe in their Gen 6 and Gen 7 FC HBAs including their latest LPe3500x series of FC HBAs.
- Most major operating systems used in modern data centers have added in-box support for FC-NVMe.
- Most recently, Pure Storage added support for FC-NVMe as a front-end transport for FlashArray //X R2, //X R3, and //C array attachments to the local FC SAN. Beginning with Purity//FA 6.1.0, FC-NVMe and FC-SCSI protocols can exist on different ports on the same storage array—providing users the ability to effortlessly transition to an end-to-end FC-NVMe solution.

All the pieces are now in place. We tested real-world database applications over a FC SAN, comparing the performance levels achieved with FC-SCSI to those achieved with FC-NVMe. The goal of these tests was to determine if database performance improvements are significantly observable for the database administrator when the database storage is configured using the new FC-NVMe storage protocol when compared to legacy FC-SCSI connectivity.

To produce clear observations, we set out to construct the most simplified configuration in order to isolate just those factors related to the I/O protocol used between the server and storage without adding additional elements that could cloud the database performance improvements. The test used the following configuration:

- A single database server, SAN connected using the latest Emulex LPe35002 Gen 7 HBA to a Brocade Gen 7 G720 switch at 32G.
- A Pure Storage FlashArray //X50 R2 array also connected to the SAN via a 32G port connection.
- A storage volume provisioned on the array and mapped to the server. The volume was partitioned and mounted as a single XFS disk to be used as the storage volume for the database instance.
- A database scaled in size so that the size of the dataset is 10x the size of available server memory.

The objective here was to ensure that a sufficient amount of the database would not be cached on the host ensuring that benchmark queries would require significant disk I/O to fulfill the benchmark query requests. We found that this storage-to-memory scale factor accurately reflected how most customers utilize their SAN resources for enterprise databases. The database benchmark was performed first with the storage connection configured for FC-NVMe and then compared with the storage connection configured for traditional FC-SCSI.

For database load testing, we used a test tool called HammerDB. HammerDB is an open-source utility hosted on GitHub by the TPC Council. HammerDB is the leading benchmarking and load testing software tool for the world's most popular databases supporting Oracle Database, Microsoft SQL Server, IBM Db2, PostgreSQL, MySQL, and MariaDB. For our testing, we utilized HammerDB's TPROC-C profile. While not the same as the full TPC TPC-C official benchmark, it is a proven methodology for predicting official TPC results. Our testing was only to evaluate I/O performance and not to determine database performance limits or to make database comparisons.

Single-Server Oracle Database 19c Testing

We constructed a single-node Oracle Database 19c instance on a server running the Red Hat RHEL 8.3 operating system. This test used the following configuration:

- An industry-standard 2U x64 PCIe Gen 3 server with two Intel Xeon Platinum 8280 processors for a total of 112 logical cores, 128 GB of RAM configured with performance BIOS settings.
- A single storage volume of 1.5 TB was configured on the Pure Storage FlashArray //X50 R2 and connected via a 32G target port to the Brocade G720 Gen 7 Fibre Channel switch.
- The server was connected to the SAN fabric using an Emulex LPe35002 Gen 7 HBA running at 32G. Storage for the database was mounted using a single XFS formatted file system.
- The Oracle Database 19c instance used an 8 KB data block size and used the dedicated connection option using the SETALL (Asynchronous and Direct IO) file system I/O type option.
- A 500 GB dataset size hhich maintained the 1:10 relationship between server memory used for the database and the Oracle dataset size to create 5000 warehouses.

We used the HammerDB TPC-C profile for Oracle databases. The test process ran through ramping iterations of virtual users until the maximum transactions per minute (TPM) could be achieved. Each virtual user was set to perform 1 million transactions per user, and we concluded that maximum TPM was achieved at 1,000 virtual users. The test runs used a two-minute ramp up time followed by five minutes of data collection. The key metric used for measurement was the database

transactions per minute (TPM). TPM is the number of database user commits plus the number of user rollbacks. In synchronization, we used the Linux SAR command to capture system statistics such as CPU utilization of both user space and system processes. For CPU comparison purposes, we use the %SYS (system processes) which is the correct part of CPU utilization to compare SCSI process overhead to NVMe.

The purpose of this test was to see if the Oracle database performs this TPC-C test profile better when using traditional FC-SCSI for connecting the SAN storage or the new FC-NVMe protocol now offered on the Pure Storage FlashArray //X50 R2. We used the configuration and test processes described above to find the maximum Oracle TPM while measuring the server processor utilization when connected to the storage array using FC-NVMe and then again using FC-SCSI. No other configuration changes were made between the two test runs. We concluded that maximum Oracle TPM was achieved using FC-NVMe and was *76% greater* than the maximum TPM achieved using FC-SCSI. The system processes of the server were also lower using FC-NVMe, achieving an efficiency score that was *103% better* than that of the traditional SCSI I/O stack.

See the following figures for results and testbed configuration.

Figure 1: Single-Server Oracle Database 19c Test—Gains Achieved in TPM and CPU Efficiency

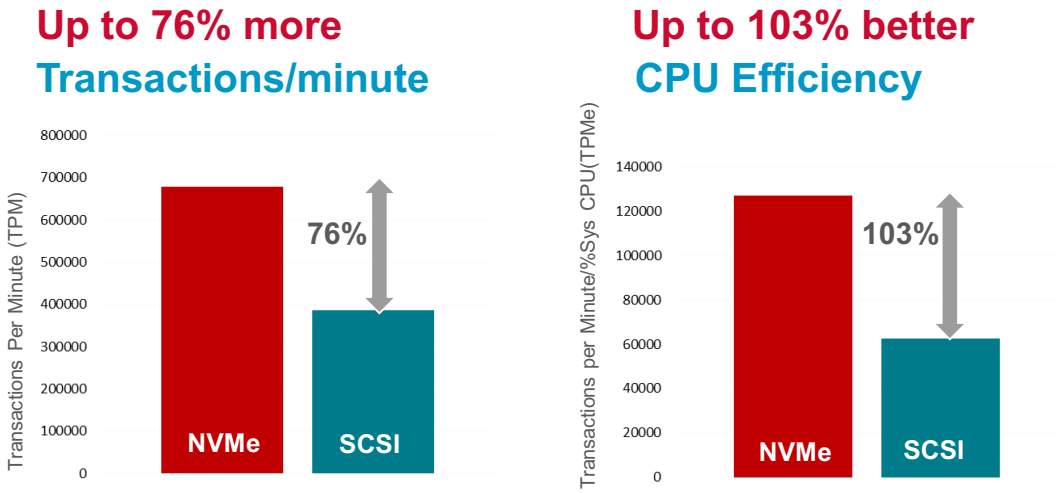
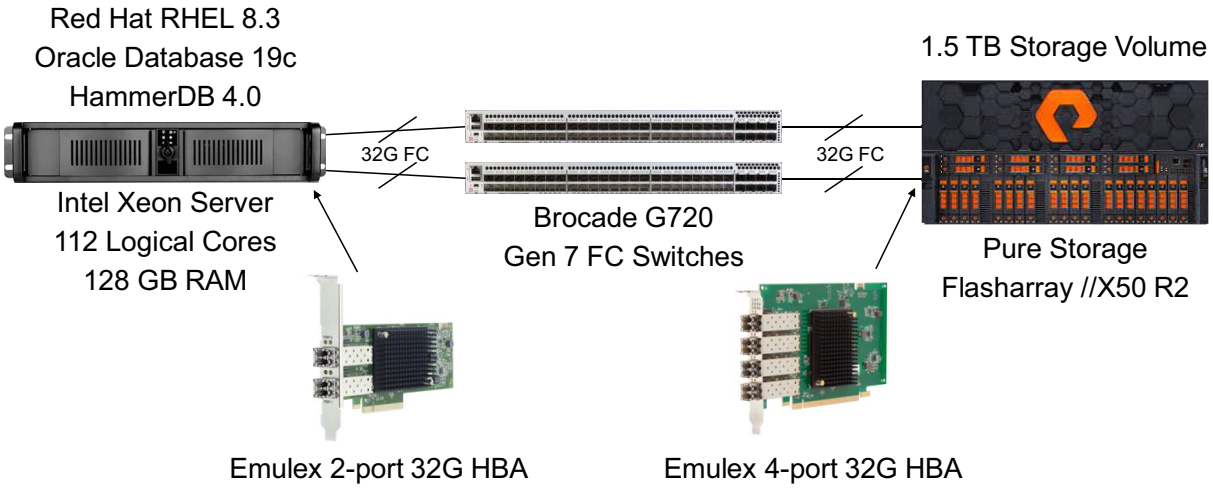


Figure 2: Single-Server Oracle Database 19c Test Environment



Single-Server Microsoft SQL Server 2019 for Linux Testing

For the second application test, we utilized Microsoft SQL Server 2019 running on Red Hat RHEL 8.3 Linux. The server configuration, HBA parameters, and storage configuration were the same as those used for the previous Oracle test. The test processes used for SQL Server are somewhat different. For this reason, direct comparison between Oracle and SQL Server should not be made and was not the objective of this exercise.

Again, we used the HammerDB TPC-C profile but used the configuration parameters designed for Microsoft SQL Server. The test iterated through varying amounts of virtual users, each performing one million transactions per user, and achieved maximum TPM at 56 virtual users. Again, we found that the maximum TPM was achieved using FC-NVMe storage connectivity. It performed *49% better* than the exact same configuration using traditional FC-SCSI storage connectivity. Again we found that the system portion of server processor utilization performed more efficiently, with *37% improved* %SYS CPU efficiency using FC-NVMe compared to FC-SCSI.

See the following figures for results and testbed configuration.

Figure 3: Single-Server Microsoft SQL Server 2019 for Linux Test—Gains Achieved in TPM and CPU Efficiency

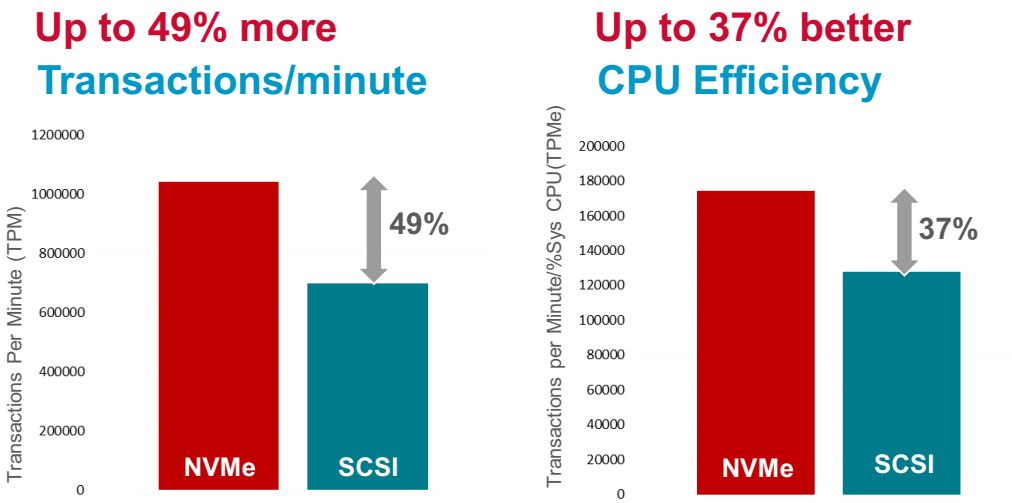
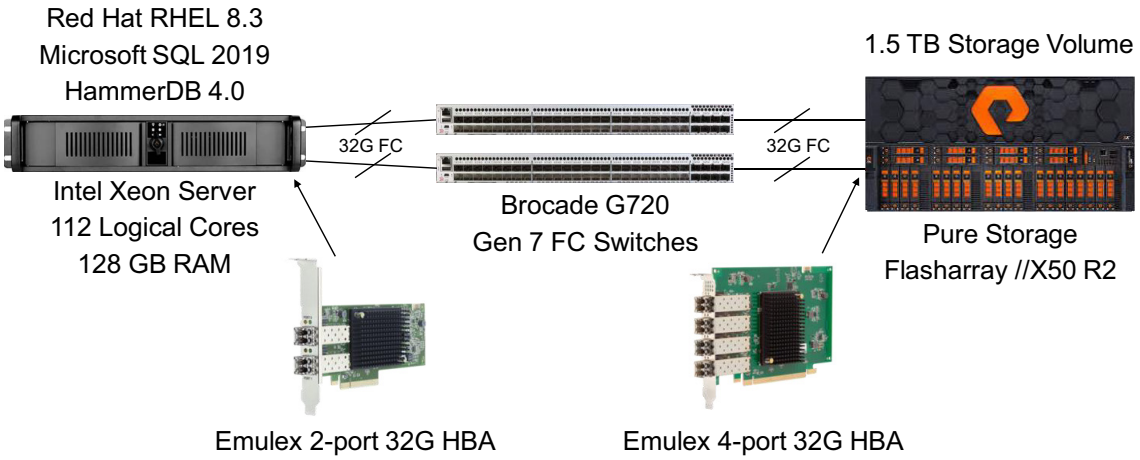


Figure 4: Single-Server Microsoft SQL Server 2019 for Linux Test Environment



Summary of Results

As mentioned at the beginning of this paper, the demands on business-critical applications continue to escalate while IT budgets shrink. The phrase *do more with less* has gone beyond a cliché to standard operating procedure. In past years, a gain of 20% in application performance justified significant IT investments by corporations. In the testing for this paper, by utilizing the NVMe over Fibre Channel protocol as opposed to using the classic SCSI over Fibre Channel protocol, the number of transactions per minute for an Oracle 19c database increased by 76% and the number of transactions per minute for a Microsoft SQL 2019 database increased by 49%. These dramatic productivity increases were accomplished on an IT footprint that may already be deployed in your environment today. If so, the cost to realize these performance gains may be zero.

Businesses depend on database applications to compete or even to survive. With the productivity gains for these core applications being so compelling, it is imperative to investigate all avenues available to capitalize on the performance gains such as those described in this paper. Connect with your local Brocade team and leverage their expertise to review your environment and define a strategy to realize these performance benefits.

Server Configuration

Component	Details
Server	2U Rackmount PCIe Gen 3
Logical CPU	112
Memory	128GB
Processor	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz
HBA	Emulex LPe35002 32G 2p FC HBA
HBA Firmware	12.8.351.22
HBA driver (lpfc)	12.8.0.1
Switch 1	Brocade G720 (FOS v9.0.0)
Switch 2	Brocade G720 (FOS v9.0.0)
OS	RHEL 8.3
Storage Array	FlashArray //X50 R2 (Purity//FA 6.1.0)
Topology tested	Single Disk - Single Path
Storage Volume	Size = 1.5T

Oracle User Settings	/etc/sysctl.conf fs.file-max = 6815744 kernel.sem = 250 32000 100 256 kernel.shmmni = 4096 kernel.shmall = 1073741824 kernel.shmmax = 4398046511104 net.core.rmem_default = 262144 net.core.rmem_max = 4194304 net.core.wmem_default = 262144 net.core.wmem_max = 1048576 fs.aio-max-nr = 3145728 net.ipv4.ip_local_port_range = 9000 65500 vm.swappiness = 1 kernel.panic_on_oops = 1 net.ipv4.conf.all.rp_filter = 2 net.ipv4.conf.default.rp_filter = 2 vm.hugetlb_shm_group = 54321 vm.panic_on_oom = 1 vm.nr_hugepages = 35000	/etc/security/limits.conf oracle soft nofile 131072 oracle hard nofile 131072 oracle soft nproc 131072 oracle hard nproc 131072 oracle soft stack 131072 oracle hard stack 131072 oracle hard memlock 134217728 oracle soft memlock 134217728 oracle soft core unlimited oracle hard core unlimited
Oracle DB Settings	filesystemio_options = SETALL DB_WRITER_PROCESSES = (112 / 4) Processes = 9000 Change TEMP to big datafile	DB block size = 8K (default) SGA = 50G PGA = 12G REDO Log file size = 1 GB x 3 Created 1TiB bigfile for TPCCTAB tablespace
Test Tool	Hammer DB version 4.0	
Hammer DB Settings	TPC-C Test Profile	Warehouses = 5000
RHEL 8.3 OS Settings	Kernel 4.18.0-240.10.1.el8_3.x86_64 transparent huge pages = disabled huge pages = allocated	File System = XFS scsi blk mq = enabled Firewall and Selinux disabled
SQL Server 2019 User Settings	/etc/sysctl.conf Default	/etc/security/limits.conf Default
SQL Database Settings Default Installation and File Layout on SAN Storage partition	DB Memory Allocation = 1:10	Dataset Size Used = 500 GB
Test Tool	Hammer DB Version 4.0	
Hammer DB Settings	TPC-C Test Profile	Warehouses = 5000
RHEL 8.3 OS Settings	Kernel 4.18.0-240.10.1.el8_3.x86_64 Transparent Huge Pages = Enabled Firewall and Selinux disabled	Transactions Per User = 1M File System = XFS scsi blk mq = enabled

BROCADE
A Broadcom Company

EMULEX®
Fibre Channel HBAs

PURESTORAGE®