Lenovo ThinkSystem DM Series Performance Guide

Introduces DM Series performance concepts

Explains ONTAP performance fundamentals

Explains controller reads and writes operations

Provides basic infrastructure check points

Vincent Kao

LENOVO PRESS

Abstract

Lenovo® ThinkSystem™ DM Series Storage Arrays run ONTAP data management software, which gives customers unified storage across block-and-file workloads. This document covers the performance principles of the ONTAP operating system and the best practice recommendations.

This paper is intended for technical specialists, sales specialists, sales engineers, and IT architects who want to learn more about the performance tuning of the ThinkSystem DM Series storage array. It is recommended that users have basic ONTAP operation knowledge.

At Lenovo Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges.

See a list of our most recent publications at the Lenovo Press web site:

http://lenovopress.com

Do you have the latest version? We update our papers from time to time, so check whether you have the latest version of this document by clicking the **Check for Updates** button on the front page of the PDF. Pressing this button will take you to a web page that will tell you if you are reading the latest version of the document and give you a link to the latest if needed. While you're there, you can also sign up to get notified via email whenever we make an update.

Contents

Introduction

Lenovo ThinkSystem DM Series Storage Arrays are unified storage systems that are designed to provide performance, simplicity, capacity, security, and high availability for businesses of any size. Powered by the ONTAP software, ThinkSystem DM Series Storage Arrays deliver hybrid and all-flash storage with enterprise-class storage management capabilities and a wide choice of host connectivity options, flexible drive configurations, and enhanced data management features.

Figure 1 Lenovo ThinkSystem DM5000H

For more information about ThinkSystem DM Series Storage, see the Lenovo Press product guide:

https://lenovopress.com/lp0941-lenovo-thinksystem-dm-series-unified-storage-arrays

ONTAP software introduction

ONTAP software unifies data management across flash, disk, and cloud to simplify your storage environment. It bridges current enterprise workloads and new emerging applications. It builds the foundation for a Data Fabric, making it easy to move your data where it is needed across flash, disk, and cloud resources.

For complete ONTAP documents, see ThinkSystem storage online help:

https://thinksystem.lenovofiles.com/help/index.jsp?topic=%2Fcom.lenovo.thinksystem.storage.doc%2Foverview_storage.html

Aggregates and RAID groups

Modern RAID technologies protect against disk failure by rebuilding a failed disk's data on a spare disk. The system compares index information on a “parity disk” with data on the remaining healthy disks to reconstruct the missing data, all without downtime or a significant performance cost.

An aggregate consists of one or more RAID groups. The RAID type of the aggregate determines the number of parity disks in the RAID group and the number of simultaneous disk failures the RAID configuration protects against.

The default RAID type, RAID-DP (RAID-double parity), requires two parity disks per RAID group and protects against data loss in the event of two disks failing at the same time. For RAID-DP, the recommended RAID group size is between 12 and 20 HDDs and between 20 and 28 SSDs.

You can spread out the overhead cost of parity disks by creating RAID groups at the higher end of the sizing recommendation. This is especially the case for SSDs, which are much more reliable than capacity drives. For HDD aggregates, you should balance the need to maximize disk storage against countervailing factors like the longer rebuild time required for larger drive size in the RAID groups.

RAID protection levels for disks

ONTAP supports three levels of RAID protection for aggregates. Your level of RAID protection determines the number of parity disks available for data recovery in the event of disk failures.

With RAID protection, if there is a data disk failure in a RAID group, ONTAP can replace the failed disk with a spare disk and use parity data to reconstruct the data of the failed disk.

- RAID4: With RAID4 protection, ONTAP can use one spare disk to replace and reconstruct the data from one failed disk within the RAID group. RAID4 option is only available through CLI.
- RAID-DP: With RAID-DP protection, ONTAP can use up to two spare disks to replace and reconstruct the data from up to two simultaneously failed disks within the RAID group.
- RAID-TEC: With RAID-TEC protection, ONTAP can use up to three spare disks to replace and reconstruct the data from up to three simultaneously failed disks within the RAID group.

Default RAID policies for aggregates

Either RAID-DP or RAID-TEC is the default RAID policy for all new aggregates. The RAID policy determines the parity protection you have in the event of a disk failure.

RAID-DP provides double-parity protection in the event of a single or double disk failure. RAID-DP is the default RAID policy for the following aggregate types:

- All flash aggregates
- Flash Pool aggregates
- Enterprise hard disk drive (HDD) aggregates

RAID-TEC is supported on all disk types and all platforms, including all-flash arrays. Aggregates that contain larger disks have a higher possibility of concurrent disk failures. RAID-TEC helps to mitigate this risk by providing triple-parity protection so that your data can survive up to three simultaneous disk failures. RAID-TEC is the default RAID policy for capacity HDD aggregates with disks that are 6 TB or larger.

Considerations for sizing RAID groups

Configuring an optimum RAID group size requires a trade-off of factors. You must decide which factors—speed of RAID rebuild, assurance against risk of data loss due to drive failure, optimizing I/O performance, and maximizing data storage space—are most important for the aggregate that you are configuring.

When you create larger RAID groups, you maximize the space available for data storage for the same amount of storage used for parity (also known as the “parity tax”). When a larger disk fails in a RAID group, reconstruction time is increased, impacting performance for a longer period of time. In addition, having more disks in a RAID group increases the probability of a multiple disk failure within the same RAID group.

HDD or array LUN RAID groups

You should follow these guidelines when sizing your RAID groups composed of HDDs or array LUNs:

- All RAID groups in an aggregate should have a similar number of disks.

  The RAID groups do not have to be exactly the same size, but you should avoid having any RAID group that is less than one half the size of other RAID groups in the same aggregate when possible.

- The recommended range of RAID group size is between 12 and 20.

  The reliability of enterprise hard disk drives can support a RAID group size of up to 28, if needed.

- If you can satisfy the first two guidelines with multiple RAID group sizes, you should choose the larger size.

SSD RAID groups in Flash Pool aggregates

The SSD RAID group size can be different from the RAID group size for the HDD RAID groups in a Flash Pool aggregate. Usually, you should ensure that you have only one SSD RAID group for a Flash Pool aggregate, to minimize the number of SSDs required for parity.

SSD RAID groups in SSD aggregates

You should follow these guidelines when sizing your RAID groups composed of SSDs:

- All RAID groups in an aggregate should have a similar number of drives.

  The RAID groups do not have to be exactly the same size, but you should avoid having any RAID group that is less than one half the size of other RAID groups in the same aggregate when possible.

- For RAID-DP, the recommended range of RAID group size is between 20 and 28.

Introduction to ONTAP performance

The fundamental unit of work performed by storage systems is a *data operation* (typically shortened to simply *op*) that either reads or writes data to or from storage systems. There are other types of operations, especially in NFS and CIFS/SMB environments, operations such as creation/deletion of files and directories, lookups, and get and set attributes. Our discussion focuses primarily on read and write ops.

The complexities surrounding performance come from the many variables that affect performance. In addition, there are many different types of derived measurements describing performance called metrics. Among these metrics, two are considered most significant and believed to accurately characterize performance at its highest level:

- Throughput
- Latency

The first, throughput, describes the amount of work the system is doing by expressing units of work over time: for example, megabytes per second (MBps) or input/output operations per second (IOPS). The second, latency, describes the time it takes to complete a unit of work: for example, a user read or write operation expressed in milliseconds per operation (ms/op) units.

On all-flash arrays, latency is expressed in microseconds per operation, due to flash's very much higher performance. Note that the term latency in the context of this document is

functionally equivalent to round trip response time. This terminology, though technically questionable, is a long-standing tradition in the storage industry and would be prohibitive to change. Tens of thousands or hundreds of thousands of operations take place every second, so throughput and latency are typically expressed as averages normalized over a given time range (for example, per second) and a unit of work (for example, per operation).

The ONTAP operating system and the underlying cluster hardware work efficiently to make sure data is secure, reliable, and always available. Collectively, the operations mix generated by applications is uniquely referred to as an *application set workload*, often shortened to simply *workload*.

Workload characteristics that can affect and be used to describe performance include:

- Throughput. The number of operations or amount of data payload over a given period.
- Concurrency. The number of operations in flight (or resident) at a given point in time.
- Operation size. The size of the operations requested. The data portion of the operation often referred to as block size or payload.
- Operation type. The type of operation requested of the storage system (for example, read, write).
- Randomness. The distribution of data access across a dataset in an unpredictable pattern.
- Sequentiality. The distribution of data access across a dataset in a repeatable pattern. Many patterns can be detected: forward, backward, skip counts, and others.
- Working set size. The amount of data considered to be active and frequently used to complete work.
- Dataset size. The amount of data that exists in a system that is both active and at rest.

Varying any of these workload characteristics ultimately ends up affecting the performance of the system and can be observed through measured changes in either latency or throughput. In many production environments, application workload almost always increases over time, often without warning. Therefore, the performance of the storage system must be known.

With this knowledge, plans to allocate more resources or rebalance workloads can be made to meet the demands placed upon the system.

Performance relationships

There are some guiding principles behind performance that can be useful in day-to-day operations. These can be stated as relationships between the fundamental characteristics of a workload and their impact on performance:

- Throughput is a function of latency.
- Latency is a function of throughput.
- Latency is a function of service times and wait times. Wait times make up most the time and are a function of utilization, which is a function of load.
- Throughput is a function of concurrency, operation size, and randomness of operations or access patterns.
- Host applications control the operation mix, operation size, randomness, and concurrency.

These relationships can be summarized by an exponential growth curve as depicted in Figure 2, where response time (or latency) increases nonlinearly as utilization (or throughput) increases.

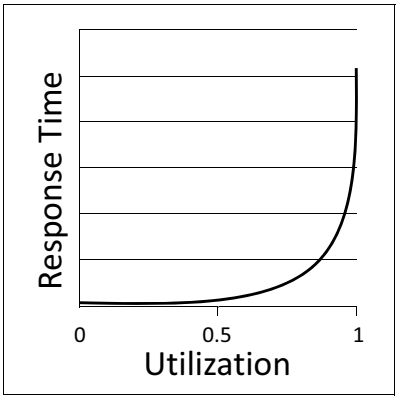

Figure 2 Response time exponential growth curve as utilization is saturated

Throughput and latency

Workloads can be defined as either closed-loop or open-loop systems. In closed-loop systems, a feedback loop exists. Subsequent operation requests from applications are dependent upon the completion of previous operations and, when bounded by the number of concurrent operation requests, limit the offered load. In this scenario, the number of concurrent requests is fixed, and the rate that operations are completed depends on how long it takes (latency) for previous operations to be completed. The SQL database access is an example of closed-loop systems. Simply put, in closed-loop systems, throughput is a function of latency; if latency increases, throughput decreases. Latency tends to be more fixed, and increasing concurrency increases throughput. Imagine the video streaming services. Single video stream is another example of closed-loop systems, since the frame sequence is fixed.

In open-loop systems, operations are performed without relying on feedback from previous operations. This configuration can be a single enterprise-class application generating multiple asynchronous requests or hundreds of independently running servers issuing a single threaded request, e.g. multiple concurrent VMware instances. This fact means that the response time from those operations doesn't affect when other operations are requested. The requests occur when necessary from the application. As offered load to the system increases, the utilization of the resources increases. As the resource utilization increases, so does operation latency. Because of this utilization increase, we can say that latency is a function of throughput in open-loop systems, although indirectly. Imagine the video streaming services, again. Multiple concurrent video stream requests from various users are considered open-loop systems.

Concurrency

Storage systems are designed to handle many operations at the same time. In fact, peak efficiency of the system can never be reached until it is processing a large enough number of operations such that there is always one waiting to be processed behind another process. Concurrency, the number of outstanding operations in flight at the same time, allows the storage system to handle the workload in the most efficient manner. The effect can be dramatic in terms of throughput results.

Concurrency is the number of parallel operations that can be performed at the same time. It is another way of describing parallelism. Most storage arrays are designed to process many

operations in parallel and are typically at their most efficient when processing multiple threads concurrently as opposed to a single operation at a time. One way to understand concurrency is to consider how much more work can be handled in each unit of time if multiple streams of work can be worked at the same time instead of having a single stream with a rather long queue. Consider a line of 10 people waiting to pay for items at a store. If a single cashier is open, then the cashier performs a total of 10 checkouts to clear the line and performs those one after the other. If instead 5 cashiers are open, then the queue for each averages 2 transactions, and all 10 complete much more rapidly, even though each individual transaction takes the same amount of time as in the single-cashier example.

Little's Law: A relationship of throughput, latency, and concurrency

Little's Law describes the observed relationship between throughput (arrival rate), latency (residence time), and concurrency (residents):

$L = A \times W$

This equation says that the concurrency of the system (L) is equal to the throughput (A) multiplied by the latency (W). This implies that for higher throughput, either concurrency would have to increase and/or latency would have to decrease. This explains why low-concurrency workloads (single-threaded workloads), even with low latencies, can have lower than expected throughput. Thus, to increase throughput with low latency requires more workloads to be added to the environment or more concurrency added to the workload.

Operation size

A similar effect on concurrency is observed with the size of operations on a system. More work, when measured in megabytes per second (MBps), can be done with larger operations than can be done with smaller operations. Each operation has fixed overhead associated with it. When the operation size (or data payload) is increased, the ratio of overhead to data is decreased, which allows more throughput over the same time. Similarly, when work depends on latency in low-concurrency workloads, a larger operation size increases the data throughput efficiency of each individual operation.

Small operations might have a slightly better latency than large operations, so the operations per second could be potentially higher, but the measured data throughput suffers with smaller operations.

Data access (random or sequential)

Data operations sent to a storage system access a logical location within a data file or LUN. This logical location is ultimately translated into an actual physical location on the permanent storage media. The order of operations and the access pattern of the data over time determine the randomness of a workload. When the logical addresses are ordered (next to one another), access patterns are considered sequential.

Sequentially read data exhibits better performance characteristics because fewer drive seeks and operations are required from permanent storage media. Solid-state drives (SSDs) exhibit a much lower impact from random access than spinning media. ONTAP is highly write-optimized. Due to the way writes are written to storage, almost all writes behave as if they are sequential writes. Thus, we see less improvement in random versus sequential writes.

Cluster-node system architecture overview

Storage systems are designed to store and retrieve large amounts of data reliably, inexpensively, and quickly. It is important to recognize that every workload interacts with the system differently, and there are many different workloads. This fact creates a technical challenge around providing the best performance for workload conditions that are largely unknown. Lenovo meets this challenge through innovative technologies combining the use of spinning disk, flash, and RAM.

A Lenovo storage system may be logically divided into three main areas when discussing performance. Those are connectivity, the system itself, and the storage subsystem. Connectivity refers to the network interface card (NIC) and host bus adapter (HBA) that attach the storage system to the clients and hosts. The system itself is the combination of CPU, memory, and NVRAM. Finally, the storage subsystem consists of the disks and Flash Cache and Flash Pool intelligent caching. Figure 3 logically represents a Lenovo hard disk or hybrid system.

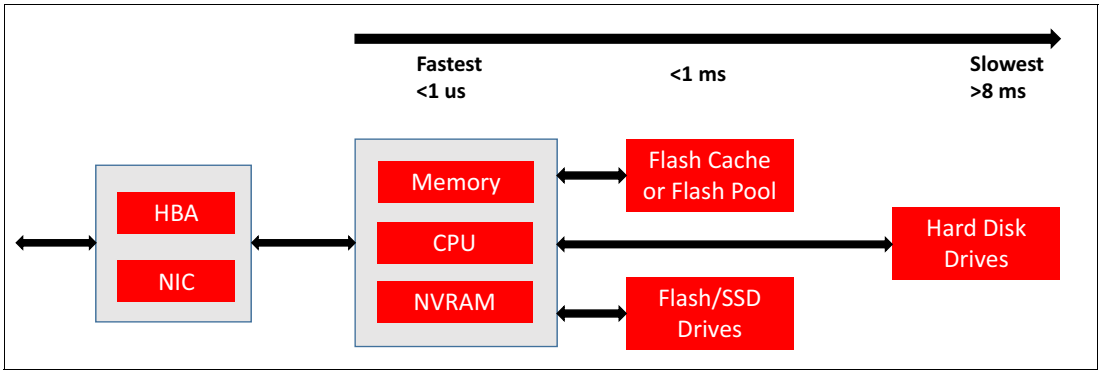

Figure 3 High-level traditional HDD or hybrid cluster node system architecture

Compare the traditional HDD or hybrid system with a Lenovo all-flash array (AFA), which is depicted in Figure 4. Notice that no spinning media are present, and there is no need for Flash Cache or Flash Pool because primary storage is very fast flash.

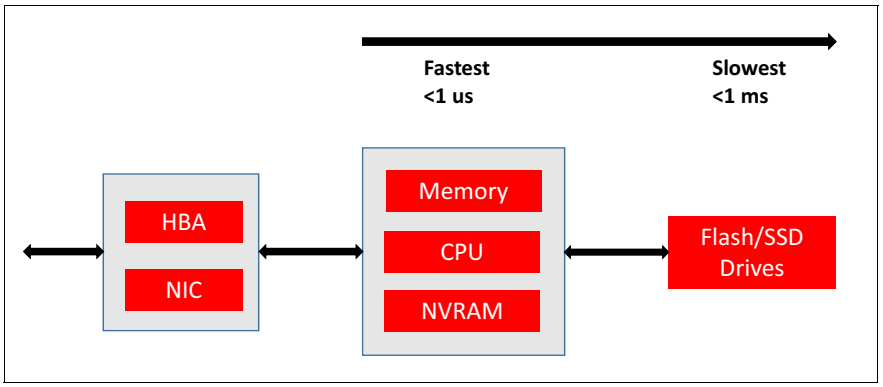

Figure 4 High-level flash/SSD node system architecture

A system running ONTAP consists of individual nodes joined together by the cluster interconnect. Every node in the cluster can store data on disks attached to it, essentially adding “copies” of the preceding architecture to the overall cluster. ONTAP has the capability to non-disruptively add additional nodes to the system to scale both system performance and capacity. An ONTAP cluster can scale both vertically and horizontally to meet the needs of the customer’s application environment.

Connectivity: NICs and HBAs

NICs and HBAs provide the connectivity to client, management, and cluster interconnect networks. Adding more or increasing the speed of NICs or HBAs can scale client network bandwidth.

Controller subsystem: memory, CPU, and NVRAM

The number of CPU cores and the amount of memory vary based on controller model. As with any computer, the CPU provides the processing power to complete operations for the system. In addition to holding the ONTAP operating system, the memory in a DM Series controller also acts as a cache. Incoming writes are staged in main memory prior to being written to disk. Memory is also used as a read cache to provide extremely fast access time to recently read data.

DM Series arrays also contain NVRAM. NVRAM is battery-backed memory used to protect inbound writes as they arrive. This fact allows write operations to be safely acknowledged without having to wait for a disk operation to complete, greatly reducing write latency. High-availability (HA) pairs are formed by mirroring NVRAM across two controllers. By staging writes in memory and NVRAM and then committing them to disk during a consistency point (CP), DM Series can both acknowledge writes very quickly and make almost all writes appear to be sequential. This is because at a CP the storage controller optimizes all the writes stored in memory and writes long stripes to disk.

Increasing the capacity and performance of these components requires either upgrading to a higher performance controller model or upgrading the version of ONTAP software running on your controllers. ONTAP software upgrades typically include performance boosts where continuous code optimizations provide performance boosts that can be quite dramatic.

Storage subsystem: Disks, Flash Cache, and Flash Pool

Spinning disk drives are the slowest persistent storage media available and have traditionally been the bottleneck in storage performance. The typical response times for spinning disks range from 3ms to 5ms for 10,000RPM and 7200RPM drives, respectively. Solid-state disks are generally an order of magnitude faster and both significantly reduce the latency at the storage subsystem and change the nature of performance tuning and sizing. Ultimately, the type of disk needed for a specific application depends on capacity, performance requirements, and workload characteristics.

Generally, when sizing or tuning a storage system design to optimize performance with spinning disks, high-performance designs call for maximizing the number of drive spindles being used to spread I/O across large numbers of disks. This allows the storage array being configured to parallelize I/O across large numbers of disks. In addition to maximizing the number of disks in the array, the other principal method of increasing performance is to add faster media, either RAM or SSDs, to act as an intermediate cache holding hot data so that repeated access can be served from cache rather than requiring much more latency-intensive disk operations.

Flash Cache and Flash Pool technology leverage the performance of solid-state flash technology with the capacity of spinning media. Flash Cache typically operates as an additional layer of read cache for the entire system. It caches recently read, or “hot,” data for future reads. Flash Pool serves as a read cache in a fashion similar to that of Flash Cache at the aggregate level as opposed to the system level. This fact allows improved cache

provisioning for specific workloads. Flash Pool also caches random overwrites, improving write latency as well.

The wide availability and rapidly falling prices of SSDs have changed this paradigm. Now when designing, sizing, or tuning high-performance arrays, you would choose an all-flash array. Performance requirements can now be satisfied by SSDs and therefore rarely rely on additional RAM or caches to store hot data because the typical SSD is an order of magnitude more highly performing than even the fastest 10k spinning disks. The other side effect of the huge speed increases with SSDs is that it's no longer required to spread high-performance workloads across a large number of spindles to achieve very high performance required. We now frequently see 40 to 50 spindles collapsed to a couple of SSDs that can support the IOPS requirements that might have taken 40 to 50 HDDs, even when the space provided by all those drives wasn't necessary. Of course, rapid increases in SSD capacities are also leaving HDDs behind.

With the advent of all-flash arrays, all storage is flash, and therefore main storage is an order of magnitude more highly performing. It doesn't require similar caching strategies and moves the performance bottleneck from the SSDs themselves to the controller and CPU.

Data storage and retrieval

The fundamental purpose of a storage system is to provide services to access data reliably (without error), persistently (always available), securely, and quickly. A DM Series does this through presenting storage abstractions, such as volumes, LUNs, and file systems, that are physically hosted on a pool of resources referred to as a cluster. Clusters are composed of individual nodes connected through a back-end cluster interconnect network. Every node is an autonomous system managing its dedicated resources running technologically advanced software that flawlessly orchestrates these services called ONTAP.

DM Series read operations

Read operations can be serviced from memory, flash-based cache, or disk (which may be either spinning or flash-based drives). The workload characteristics and capabilities of the system determine where reads are serviced and how quickly. Knowing where reads are serviced can help set expectations as to the overall performance of the system. In the following diagrams, components and links in red highlight the activity described.

In the slowest case (Figure 5), read requests that are not cached anywhere must come from disk. After being read from disk, the data is kept in main memory.

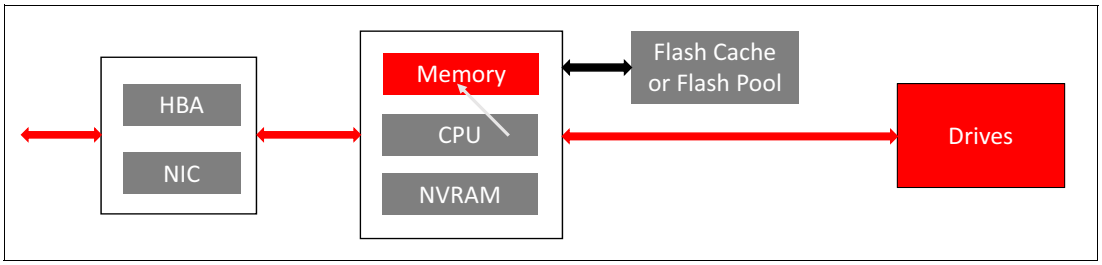

Figure 5 Read from disk

If this data is read again soon, it is possible for the data to still be cached in main memory, making subsequent access extremely fast because no disk access would be required (Figure 6).

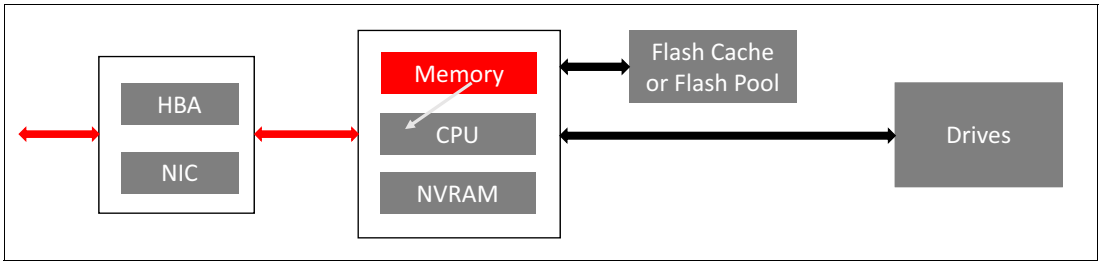

Figure 6 Read from memory

When more room is needed in the main memory cache, as is common with working sets larger than the memory cache, data is evicted. If Flash Cache or Flash Pool is in the system, that block could be inserted into the flash-based cache. In general, only randomly read data and metadata are inserted into flash-based caches (Figure 7).

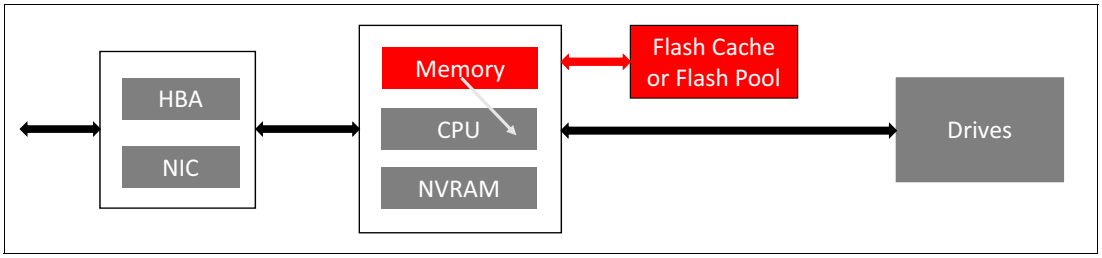

Figure 7 Write to flash

After data is inserted into Flash Cache, subsequent reads of this block unable to be serviced from the memory cache would be served from the flash-based cache (Figure 8) until they are evicted from the flash-based cache. Flash access times are significantly faster than those of disk, and adding cache in random read-intensive workloads can reduce read latency dramatically. Of course, on all-flash arrays the access times from the drives are greatly reduced. Flash arrays generally don't use intermediate Flash Cache because they don't tend to accelerate access over the already very fast flash drives being used for storage.

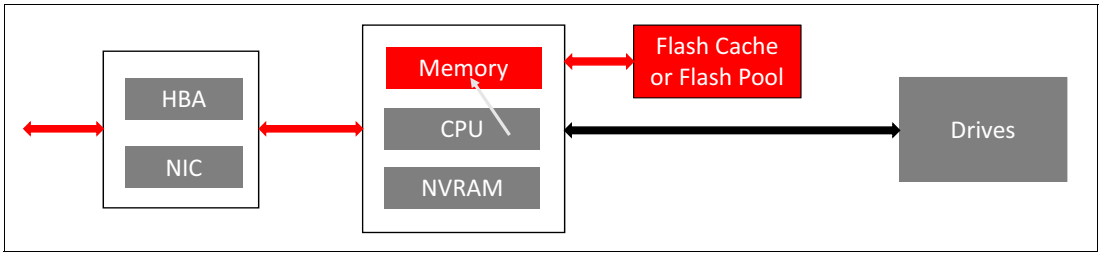

Figure 8 Read from flash

Incoming reads are continually being checked for access patterns. For some data access patterns, such as sequential access, ONTAP predicts which blocks a client might want to access prior to the client ever requesting. This "read-ahead" mechanism preemptively reads blocks off disk and caches them in main memory. These read operations are serviced at faster RAM speeds instead of waiting for disk when the read request is received. Even with vastly faster flash drives, data residing in a memory cache is still faster than going to the disk for the same data.

DM Series write operations

For most storage systems, writes must be placed into a persistent and stable location prior to acknowledging to the client or host that the write was successful. Waiting for the storage system to write an operation to disk for every write could introduce significant latency. To solve this problem, DM Series systems use battery-backed RAM to create nonvolatile RAM (NVRAM) to log incoming writes.

NVRAM is divided in half, and only one half is used at a time to log incoming writes. When controllers are in highly available pairs, half of the NVRAM is used to mirror the remote partner node's log, while the other half is used for logging local writes. The part that is used for logging locally is still split in half, just like a single node as shown in Figure 9.

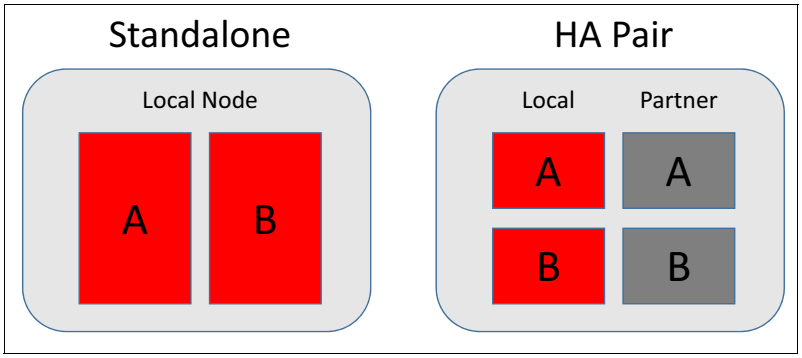

Figure 9 NVRAM segmenting: standalone and HA pair

When a write enters a DM Series, the write is logged into NVRAM and is buffered in main memory. After the data is logged in persistent NVRAM, the write is acknowledged to the client (Figure 10). NVRAM is accessed only in the event of a failure.

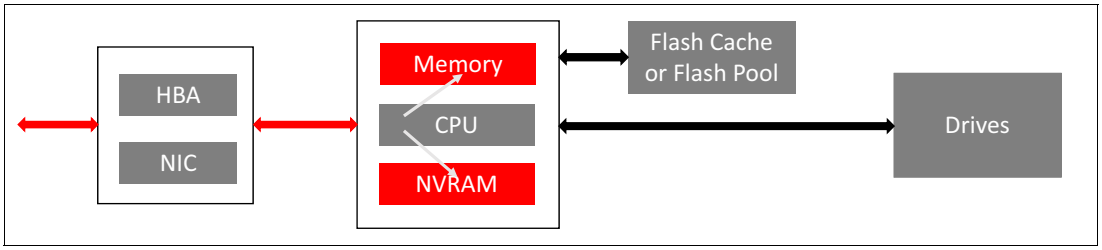

Figure 10 Accepting a write

At a later point in time, called a consistency point (CP), the data buffered in main memory is efficiently striped to disk (Figure 11). CPs can be triggered for several reasons, including time passage, NVRAM utilization, or system-triggered events such as a Snapshot copy.

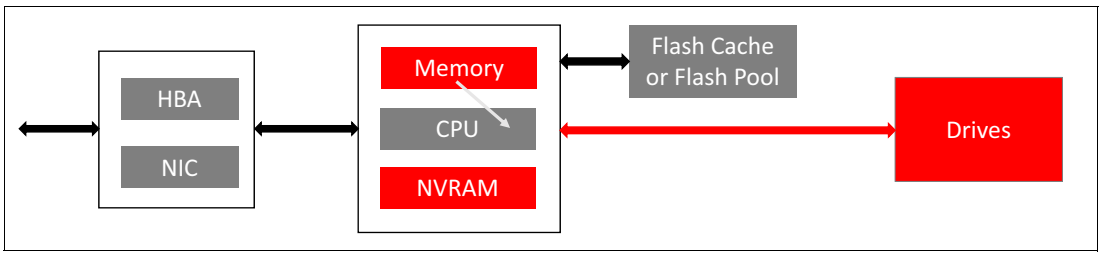

Figure 11 Consistency point

In general, writes take a minimal amount of time, on the order of low milliseconds to sub-milliseconds. If the disk subsystem is unable to keep up with the client workload and becomes too busy, write latency can begin to increase.

When writes arrive too quickly for the provisioned back-end storage, both sides of the NVRAM can fill up and lead to a scenario called a *back-to-back CP*. This fact means that both segments of NVRAM log are full, a CP is currently occurring, and another CP immediately follows the current CP's completion. This scenario affects performance because the system can't immediately acknowledge the write because NVRAM is full, and the client must wait until the operation can be logged.

Improving the storage subsystem often alleviates the back-to-back CP scenario. Increasing the number of disks, moving some of the workloads to other nodes, and considering flash-based caching or adding flash-based drives can help solve write performance issues.

The per-aggregate consistency points (PACPs), which can also reduce the incidence of back-to-back CPs because rather than having a single global CP that is ultimately only as fast as the slowest (and/or busiest) disk subsystem, the PACPs occur on a per-aggregate basis and therefore are performed on like disk types.

Performing basic infrastructure checks

Some basic diagnostic checks of your infrastructure will help you rule out obvious sources of performance problems. You should review protocol and network settings and check disk throughput and latency. If you are replicating data, you will want to monitor throughput and latency between nodes.

For the complete section in Performance Management Power Guide, see ThinkSystem Documentation Information Center:

https://thinksystem.lenovofiles.com/help/topic/performance_management_power_guide/9E17B1E2-EDE7-4CE6-BE7A-52D331E672A2_.html

The content is also available as a PDF:

https://thinksystem.lenovofiles.com/help/topic/performance_management_power_guide/M_E368AB74-F212-44B7-B18F-83FA7725CC7F_.pdf

For more information about ThinkSystem DM Series Storage, see the Lenovo Press product guide:

https://lenovopress.com/lp0941-lenovo-thinksystem-dm-series-unified-storage-arrays

The benefits of using NVMe

Non-volatile Memory Express (NVMe) technology provides higher IOPS and reduced latency from the software stack. SAS SSD flash technology has already had a huge impact on the performance of enterprise applications . However, the speed gains from SAS SSDs are approaching the limit because of bottlenecks from the protocols.

Both mainstream networking protocols, Fibre Channel Protocol (FCP) and Ethernet, use the SCSI command set for the storage protocol. In contrast, the NVMe software stack is simplified and optimized compared to SCSI. The NVMe command set takes fewer clock cycles per I/O.

Some products on the market offer NVMe support on the host side but keeping the back end with SAS SSDs. The back-end SAS SSDs run on the SCSI protocol, which is slower than NVMe protocol for flash-based drives. ThinkSystem DM7100F uses NVMe SSDs on the back end to offer end-to-end NVMe support which results in faster storage, further removing bottlenecks from the storage subsystem.

In addition, SCSI puts I/O requests into a single queue, containing a maximum queue depth (QD) of 256 commands. When the I/O requests arrive, they must wait in line while other requests are completed. Solid-state drives benefit from parallel command queues. NVMe uses PCIe bus, which supports 64K queues and each with a QD of 64K commands.

Although all applications benefit from low latency, the NVMe performance boost is especially valuable in the enterprise database applications that are sensitive to latency, such as Microsoft SQL Server, SAP HANA and Oracle. NVMe accelerates many modern workloads, including artificial intelligence (AI), machine learning (ML)/deep learning (DL) and internet of things (IoT).

Why NVMe over Fibre Channel

NVMe defines access protocols and architectures for connecting local nonvolatile storage to computers or servers. NVMe over Fabrics (NVMe-oF) increases the scalability of the NVMe interface. NVMe-oF defines how NVMe uses existing transport technologies such as FC, RoCE (RDMA over Converged Ethernet), InfiniBand and iWARP. It transports the NVMe protocol over distances and enable the use of networking switches and routers.

NVMe/FC in traditional FCP infrastructure

NVMe over Fibre Channel (NVMe/FC) is one of the NVMe-oF implementations. It runs on the same Fibre Channel as FCP.

NVMe-oF brings NVMe protocol to the SAN marketplace. Many enterprise SANs use FCP for the speed and robustness of Fibre Channel. Therefore, NVMe/FC is a reasonable choice to transition from traditional FCP to NVMe data fabrics. NVMe/FC can use the same data network component that FC uses for SCSI access to storage. It can coexist on the same host, fabrics and storage that are using FCP, enabling a seamless transition to the new technology.

Differences between NVMe/FC and FCP

NVMe/FC looks very much like FCP, which encapsulates SCSI commands inside FC frames. The reason both look similar is that NVMe/FC swaps out the SCSI commands for the streamlined NVMe command set. This simple replacement improves the throughput and latency.

NVMe adds some new names for some common structures. Table 1 maps some common structures that have different names than those used in FCP.

Table 1 FCP and NVMe/FC terms

Fibre Channel Protocol (FCP)	**NVMe/FC**
World-wide Port Name (WWPN)	NVMe Qualified Name (NQN)
LUN	Namespace

Fibre Channel Protocol (FCP)	NVMe/FC
Igroup, LUN mapping, and LUN masking	Subsystem
Asymmetric Logical Unit Access (ALUA)	Asymmetric Namespace Access (ANA)

The NVMe/FC terms have the following meaning:

- An NVMe Qualified Name (NQN) identifies an endpoint and is similar to the World-wide Port Name (WWPN) in both format (domain registration date, domain registered, and a serial number).
- A namespace ID is an identifier used by a controller to provide access to a namespace. This is nearly equivalent to a logical unit number (LUN) in SCSI. The accessibility of a volume by a host is configured from the management interfaces, along with setting the namespace ID for that host or host group. As with SCSI, a logical volume can be mapped to only a single host group at a time, and a given host group cannot have any duplicate namespace IDs.
- A subsystem is analogous to an initiator group (igroup), and it is used to mask an initiator so that it can see and mount a LUN or namespace.
- Asymmetric Namespace Access (ANA) is a new protocol feature for monitoring and communicating path states to the host operating system's Multipath I/O (MPIO) or multipath stack, which uses information communicated through ANA to select and manage multiple paths between the initiator and target.

Multipathing and failover

ONTAP supports Asymmetric Namespace Access (ANA) as part of the NVMe/FC target. Like Asymmetric Logical Unit Access (ALUA), ANA uses both an initiator-side and target-side implementation for it to be able to provide all the path and path state information that the host-side multipathing implementation to work with a storage high availability (HA) multipathing software used with each OS stack.

ANA requires both the target and initiator to implement and support ANA to function. If either side is not available or implemented, ANA isn't able to function, and NVMe/FC will fall back to not supporting storage HA. In those circumstances, applications will have to support HA for redundancy.

NVMe/FC relies on the ANA protocol to provide multipathing and path management necessary for both path and target failover. The ANA protocol defines how the NVMe subsystem communicates path and subsystem errors back to the host so that the host can manage paths and failover from one path to another. ANA fills the same role in NVMe/FC that ALUA does for both FCP and iSCSI protocols.

For multipath management, ANA in NVMe/FC is similar to DM-Multipath in SCSI. Red Hat Enterprise Linux (RHEL) 8.1 or later and SUSE Linux Enterprise Server (SLES) 15 SP1 or later are the only enterprise host OSs supporting NVMe/FC with ANA at the time of writing. Since NVMe/FC is under rapid development, the supported OS list would be updated frequently.

For the current supported OSs, see the Operating systems section of the DM7100F product guide:

https://lenovopress.com/lp1271-thinksystem-dm7100f-unified-all-flash-storage-array#operating-systems

ThinkSystem DM7100F NVMe support

Powered by the ONTAP software, ThinkSystem DM7100F delivers enterprise-class storage management capabilities with a wide choice of host connectivity options, flexible drive configurations, and enhanced data management features, including end-to-end NVMe support (NVMe/FC and NVMe drives).

NVMe/FC infrastructure shares the same hardware with FCP as discussed in the previous section. NVMe/FC and FCP can coexist in the production environment. ThinkSystem DM7100F allows the SAN users to upgrade from FCP to NVMe/FC with the existing Fibre Channel infrastructure and to protect Fibre Channel investment.

For more information about ThinkSystem DM7100F Storage Array, see the Lenovo Press product guide:

https://lenovopress.com/lp1271-thinksystem-dm7100f-unified-all-flash-storage-array

Best practices for NVMe/FC

This section lists recommendations and best practices when implementing NVMe/FC.

Fabric and switch configuration and operational best practices

Since NVMe/FC uses FC as a transport, the items in performing basic infrastructure checks section apply to NVMe/FC. NVMe/FC does not require any special configurations or best practices that differ from the general ThinkSystem DB Series FC switch best practices.

- Single-initiator zoning is a best practice
- Use WWPNs to assign zone memberships (instead of switch-port-based zone memberships or hard zoning).

Pathing

To avoid any interface single points of failure, Lenovo strongly recommends that you provision at least two paths per SVM, per node, per fabric.

NVMe added the ANA protocol to manage communicating, alerting, and managing pathing and path state changes. ANA consists of two components:

- A host-side implementation that is responsible for querying the target (ONTAP node) for current path state information.
- A storage node implementation that is responsible for alerting when there is a path state change and answering initiator-side queries for enumerations of all available paths.

The host-side ANA implementation is responsible for passing all pathing information it receives to the host's multipathing stack. The hosts multipathing stack, for instance dm-multipath, then manages path preferences and usage.

While ANA resembles ALUA in many respects, it also has some noteworthy differences. One significant difference is that indirect paths in ALUA are active nonoptimized. This means that these paths can be used but that they will use the cluster interconnect in the absence of a direct path. In ANA indirect paths are labeled Inactive, rather than Nonoptimized. This means that with ANA indirect paths aren't used and won't be used except after a takeover or giveback, which changes which paths are in direct vs. indirect. Because of this change of

behavior, it is a strong best practice recommendation that storage admins configure at least two paths from each node to each fabric per SVM. This best practice is recommended because it remediates the single point of failure that a single active path represents given that all other paths through other controllers will be inactive and won't be useable until and if there is a storage failover.

Setup and configuration

Before setting up NVMe/FC, make sure the following requirements are in place:

1. Verify that your configuration exactly matches a qualified configuration listed in the Lenovo Interoperability Matrix. Failure to do so is likely to lead to suboptimal, poorly configured storage implementation.

   The link to the latest Interop Matrix can be found at the following page:

   https://lenovopress.com/lp0584-lenovo-storage-interoperability-links

2. Enable N_Port ID virtualization (NPIV) on all fabric switches.
3. Use single-initiator zoning and use WWPNs to specify zone membership. Do not use switch port connectivity to denote zone membership or hard zoning.
4. Create NVMe/FC objects (SVMs, volumes, namespaces, subsystems, and LIFs) by using ONTAP. For detail, see the following Information Center documentation page:

   https://thinksystem.lenovofiles.com/storage/help/topic/san_administration_guide/247E9BB5-B71B-414F-B91C-CB055BA119EE_.html?cp=3_9_6_2

5. Use Lenovo ThinkSystem Intelligent Monitoring Unified Manager to monitor the health and performance of newly created NVMe objects and create reporting thresholds and alerts.

A case study in switching from FCP to NVMe/FC

To demonstrate the advantages of using NVMe/FC, HammerDB was run on both FCP and NVMe/FC solutions to compare the performance. HammerDB is the database benchmarking tool supported in Linux and Windows database environments. Lenovo provided the hardware solutions in collaboration with Broadcom.

Configuration summary:

- Lenovo ThinkSystem DM7100F
  Product Guide https://lenovopress.com/lp1271
- Lenovo ThinkSystem DB620S 32Gb FC SAN Switch
  Product guide: https://lenovopress.com/lp0580
- ThinkSystem Emulex LPe35002 32Gb FC Adapter
  Product guide: https://lenovopress.com/lp1178
- Lenovo ThinkSystem SR650 Server
- Oracle 19c Database or Microsoft SQL Server 2019 on Linux

The configuration is shown in Figure 12 on page 19.

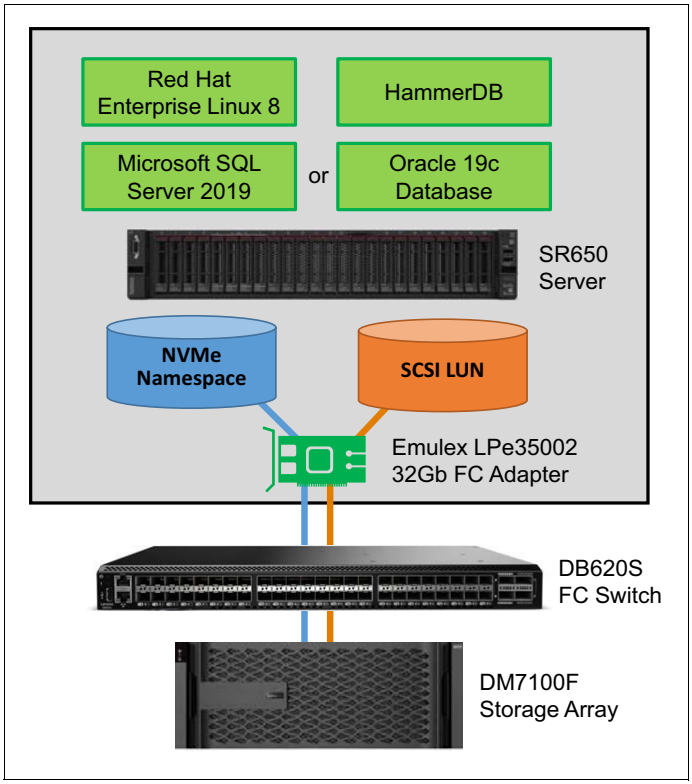

Figure 12 Case Study configuration

The results of the comparison using Oracle 19c are shown in Figure 13.

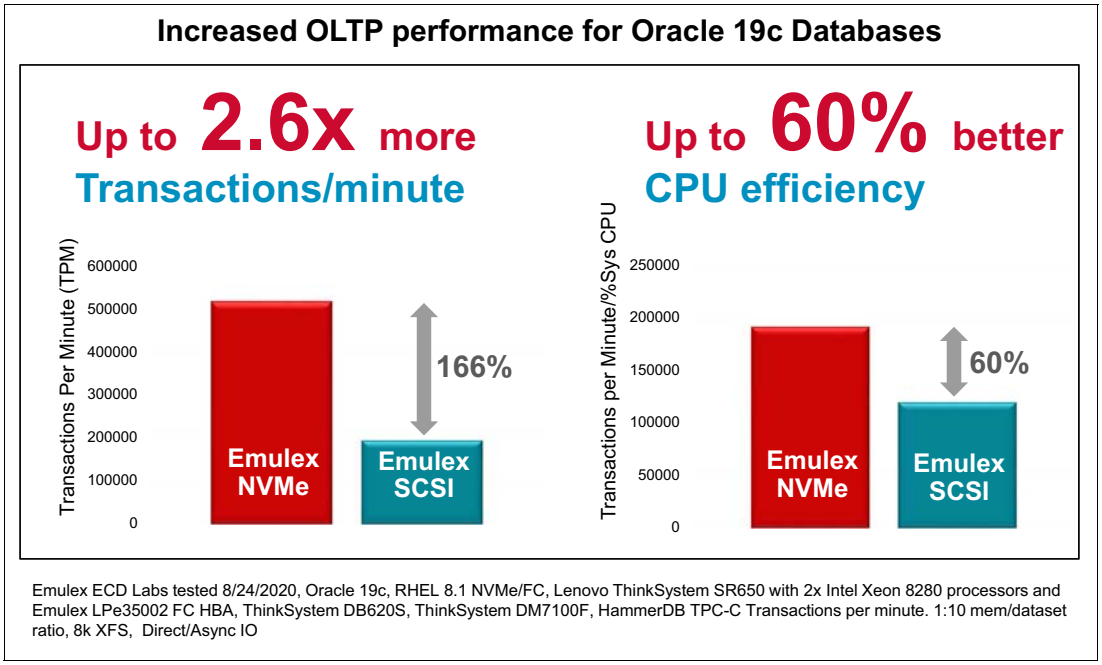

Figure 13 Oracle 19c database

The results of the comparison using Microsoft SQL Server 2019 are shown in Figure 14.

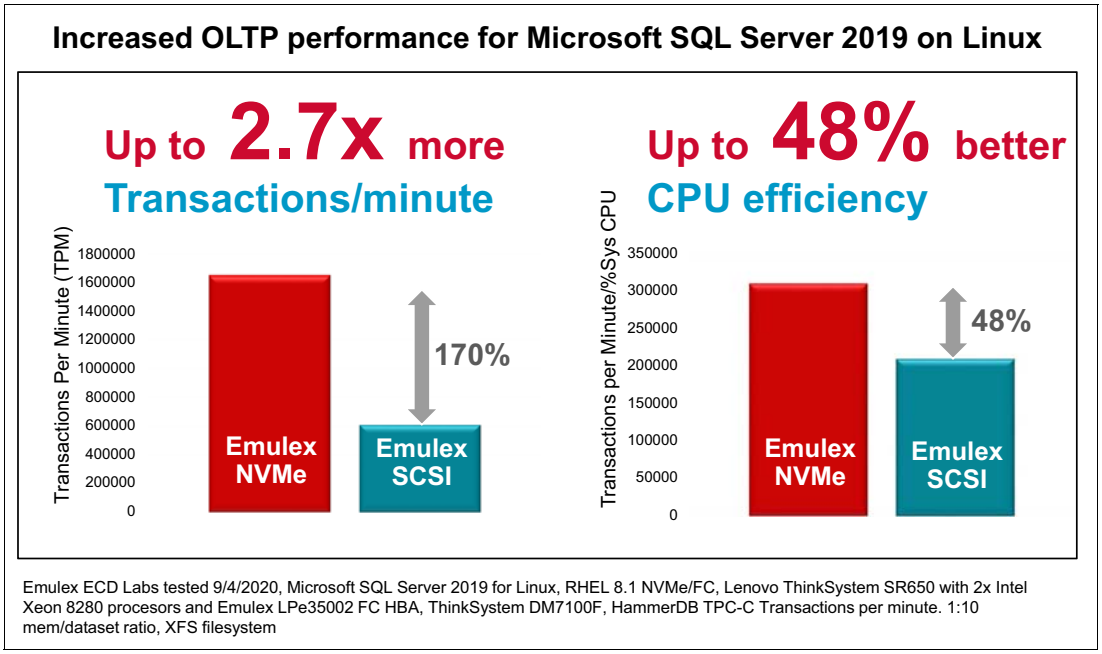

Figure 14 Microsoft SQL Server 2019 database

Figure 13 and Figure 14 show the Transactions Per Minute (TPM) and TPM per percentage of system CPU (CPU efficiency) increase when switching from FCP to NVMe/FC. The quantity of volume, multipath configuration, database settings and HammerDB workload settings are the same between FCP and NVMe/FC setups.

The end-to-end NVMe/FC solution is a clear winner over FCP.

Author

Vincent Kao is a Performance Engineer on the Lenovo Storage Development Team, based in Taipei. He is responsible for the performance analysis of RAID storage systems. Vincent earned a Master's Degree in Electrical Engineering from San Jose State University, CA and a Bachelor's Degree in Electrical Engineering from National Central University, Taiwan.

Thanks to the following people for their contributions to this project:

- Ted Vojnovich, CTO External Storage
- Yuwen Yang, Storage Development
- Shawn Andrews, Storage Development
- David Watts, Lenovo Press

Change history

September 2023:

- Minor update: Updated step 5 of “Setup and configuration” on page 18

February 2021:

- Added “A case study in switching from FCP to NVMe/FC” on page 18

June 2020:

- Added the following new sections:
  - “The benefits of using NVMe” on page 14
  - “Why NVMe over Fibre Channel” on page 15
  - “Best practices for NVMe/FC” on page 17

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document was created or updated on September 26, 2023.

Send us your comments via the **Rate & Provide Feedback** form found at
http://lenovopress.com/lp1276

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. These and other Lenovo trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by Lenovo at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of Lenovo trademarks is available from https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

Lenovo® Lenovo(logo)® ThinkSystem™

The following terms are trademarks of other companies:

Other company, product, or service names may be trademarks or service marks of others.