KIOXIA

Performance Brief

Using NVMe® Namespaces to Increase KIOXIA CM6 Series SSD Performance

Result: Better Utilization of Unused Capacity – Excellent Random Performance

A key goal of IT administrators is to deliver fast storage device performance to end-users in support of the many applications and workloads they require. With this objective, many data center infrastructures have either transitioned to, or are transitioning to NVMe storage devices given the very fast read and write capabilities they possess. Selecting the right NVMe SSD for specific or a whole slew of application workloads is not always a simple process as user requirements can vary depending on the virtual machines (VMs) and containers for which they are deployed. User needs can also dynamically change due to workload complexities and other aspects of evolving application requirements. Given these volatilities, it can be very expensive to constantly replace NVMe SSDs to meet the varied application workload requirements so maximizing their performance is an IT objective.

To achieve even higher write performance from already speedy PCIe® 4.0 enterprise SSDs, using NVMe namespaces is a viable solution. Using namespaces can also deliver additional benefits such as better utilization of a drive's unused capacity and increased performance of random write workloads. The mantra, 'don't let stranded, unused capacity go to waste when random performance can be maximized,' is a focus of this performance brief.

Random Write SSD Performance Effect on I/O Blender Workloads

The term I/O blender refers to a mix of different workloads originating from a single application or multiple applications on a physical server within bare-metal systems or virtualized / containerized environments. VMs and containers are typically the originators of I/O blender workloads.

When an abundance of applications run simultaneously in VMs or containers, both sequential and random data input/output (I/O) streams are sent to SSDs. Any sequential I/O that exists at that point is typically mixed in with all of the other I/O streams and essentially become random read/write workloads. As multiple servers and applications process these workloads and move data at the same time, the SSD activity changes from just sequential or random read/write workloads into a large mix of random read/write I/Os - the I/O blender effect.

As almost all workloads become random mixed, an increase in random write performance can have a large impact on the I/O blender effect in virtualized and containerized environments

The I/O blender effect can come into play at any time where multiple VMs and/or containers run on a system. Even if a server is deployed for a single application, the I/O written to the drive can still be highly mixed with respect to I/O size and randomness. Today's workload paradigm is to use servers for multiple applications, not just for a single application, which is why most modern servers are deployed for virtualized or containerized environments. It is in these modern infrastructures where the mix of virtualized and containerized workloads create the I/O blender effect, and is therefore applicable to almost every server that ships today.

Addressing the I/O Blender Effect

Under mixed workloads, some I/O processes that would have typically been sequential in isolation, become random, which can increase SSD read/write activity, as well as latency (or the ability to access stored data). One method used to address the I/O blender effect involves allocating more SSD capacity for overprovisioning (OP).

Overprovisioning

Overprovisioning means that an SSD has more flash memory than its specified user capacity, also known as the OP pool. The SSD controller uses the additional capacity to perform various background functions (transparent to the host) such as flash translation layer (FTL) management, wear leveling and garbage collection (GC). GC, in particular, reclaims unused storage space which is very important for large write operations.

The OP pool is also very important for random write operations as the more random the data patterns are allows the extra OP to provide space for the controller to place new data for proper wear leveling and reduce write amplification (while handling data deletions and clean up in the background). In a data center, SSDs are rarely used for only one workload pattern. Even if the server is dedicated to a single application, other types of data can be written to a drive, such as logs or peripheral data that may be contrary to the server's application workload. As a result, almost all SSDs perform random workloads and the more write-intensive the workload is, the more OP is needed on the SSD to maintain maximum performance and efficiency.

Namespaces

Namespaces divide an NVMe SSD into logically separate and individually addressable storage spaces where each namespace has its own I/O queue. Namespaces appear as a separate SSD to the connected host who interacts with them as it would with local or shared NVMe targets. They function similarly to a partition, but at the hardware level as a separate device. Namespaces are developed at the controller level and have the included benefit of dedicated I/O queues that may provide improved Quality of Service (QoS) at a more granular level.

With the latest firmware release of KIOXIA CM6 Series PCIe 4.0 enterprise NVMe SSDs, flash memory that is not provisioned for a namespace is added back into the OP pool, which in turn, ***enables higher write performance for mixed workloads***. To validate this methodology, testing was performed using a CM6 Series 3.84 terabyte[1] (TB), 1 Drive Write Per Day[2] (DWPD) SSD, provisioned with smaller namespaces (equivalent to a CM6 Series 3.2TB 3DWPD model). As large OP pools impact performance, CM6 Series SSDs can be set to a specific performance or capacity metric desired by the end user. By using namespaces and reducing capacity, ***a 1DWPD CM6 Series SSD can perform comparably in write performance to a 3DWPD CM6 Series SSD*** as demonstrated by the test results.

Testing Methodology

To validate the performance comparison, benchmark tests were conducted by KIOXIA in a lab environment that compared the performance of three CM6 Series SSD configurations with namespace sizes across the classic four-corner performance tests and three random mixed-use tests. This included a CM6 Series SSD with 3.84TB capacity, 1DWPD and 3.84TB namespace size, a CM6 Series SSD with 3.84TB capacity, 1DWPD and a namespace adjustment to a smaller 3.20TB size, and a CM6 Series SSD with 3.20TB capacity, 3DWPD and 3.20TB namespace size to compare the smaller namespace adjustment to.

The seven performance tests were run through Flexible I/O (FIO) software[3] which is a tool that provides a broad spectrum of workload tests with results that deliver the actual raw performance of the drive itself. This included 100% sequential read/write throughput tests, 100% random read/write IOPS tests, and three mixed random IOPS tests (70%/30%, 50%/50% and 30%/70% read/write ratios). These ratios were selected as follows:

70%R / 30%W: *represents a typical VM workload*
50%R / 50%W: *represents a common database workload*
30%R / 70%W: *represents a write-intensive workload (common with log servers)*

In addition to these seven tests, 100% random write IOPS tests were performed on varying namespace capacity sizes to illustrate the random write performance gain that extra capacity in the OP pool provides. The additional namespace capacities tested included a CM6 Series SSD with 3.84TB capacity, 1DWPD and two namespace adjustments (2.56TB and 3.52TB).

A description of the test criteria, set-up, execution procedures, results and analysis are presented. The test results represent probable outcomes of three different namespace sizes and associated capacity reductions have on four-corner performance and read/write mixes (70%/30%, 50%/50% and 30%/70%). There are additional 100% random write test results of four different namespace sizes when running raw FIO workloads with a CM6 Series 3.84TB, 1DWPD SSD and equipment as outlined below.

Test Criteria

The hardware and software equipment used for the seven performance tests included:

- **Dell EMC® PowerEdge™ R7525 Server:** One (1) dual socket server with two (2) AMD EPYC™ 7552 processors, featuring 48 processing cores, 2.2 GHz frequency, and 256 gigabytes1 (GB) of DDR4
- **Operating System:** CentOS™ v8.4.2105 (Kernel 4.18.0-305.12.1.el8_4.x86_64)
- **Application:** FIO v3.19
- **Test Software:** Synthetic tests run through FIO v3.19 test software
- **Storage Devices** (Table 1)**:** One (1) KIOXIA CM6 Series PCIe 4.0 enterprise NVMe SSD with 3.84 TB capacity (1DWPD)
  One (1) KIOXIA CM6 Series PCIe 4.0 enterprise NVMe SSD with 3.2 TB capacity (3DWPD)

Set-up & Test Procedures

Set-up: The test system was configured using the hardware and software equipment outlined above. The server was configured with a CentOS v8.4 operating system and FIO v3.19 test software.

Test Procedures: The following tests were conducted on the following drive configurations:

TESTS CONDUCTED

Test	Measurement	Block Size
100% Sequential Read	Throughput	128 kilobytes[1] (KB)
100% Sequential Write	Throughput	128KB
100% Random Read	IOPS	4KB
100% Random Write	IOPS	4KB
70%R/30%W Random	IOPS	4KB
50%R/50%W Random	IOPS	4KB
30%R/70%W Random	IOPS	4KB

TESTS CONDUCTED

Product	Focus	SSD Type	Capacity Size	Namespace Size
CM6 Series	Read-intensive	Sanitize Instant Erase[4] (SIE)	3.84TB	3.84TB
CM6 Series	Read-intensive	SIE	3.84TB	3.52TB
CM6 Series	Read-intensive	SIE	3.84TB	3.20TB
CM6 Series	Read-intensive	SIE	3.84TB	2.56TB
CM6 Series	Mixed-use	SIE	3.20TB	3.20TB

NOTE: *The SIE drives used for testing have no performance differences versus CM6 Series Self-Encrypting Drives[5] (SEDs) or those without encryption, and their selection was based on test equipment availability at the time of testing.*

Utilizing FIO software, the first set of seven tests were run on a CM6 Series SSD with 3.84TB capacity, 1DWPD and 3.84TB namespace size. The results were recorded.

The second set of seven FIO tests were then run on the same CM6 Series SSD except the namespace size was changed to 3.2TB to represent the namespace size of the third SSD to be tested against - the 3DWPD CM6 Series SSD with 3.2TB capacity, 3DWPD and 3.2TB namespace size. The results for these tests were recorded.

The third set of seven FIO tests were then run on the CM6 Series SSD with 3.2TB capacity, 3DWPD and 3.2TB namespace size, and the performance that the CM6 Series SSD (3.84TB capacity, 1DWPD, 3.84TB namespace size) is trying to achieve. The results for these tests were recorded.

Additionally, a 100% random write FIO test was run on the CM6 Series SSD except the namespace size was changed to 2.56TB. The results for this test was recorded. A second 100% random write FIO test was run on the CM6 Series SSD with the namespace size changed to 3.52TB. The results for this test was also recorded.

The steps and commands used to change the respective namespace sizes include:

Step 1: Delete the namespace that currently resides on the SSD:
(1) sudo nvme detach-ns /dev/nvme1 –n 1; (2) sudo nvme delete-ns /dev/nvme1 –n 1

Step 2: Create a 3.84 TB namespace and attach it	**Create a 3.52 TB namespace and attach it***	**Create a 3.2 TB namespace and attach it***	**Create a 2.56 TB namespace and attach it***
sudo nvme create-ns /dev/nvme1 -s 7501476528 -c 7501476528 -b 512 sudo nvme attach-ns /dev/nvme1 -n1 -c1	sudo nvme create-ns /dev/nvme1 -s 6875000000 -c 6875000000 -b 512 sudo nvme attach-ns /dev/nvme1 -n1 -c1	sudo nvme create-ns /dev/nvme1 -s 6251233968 -c 6251233968 -b 512 sudo nvme attach-ns /dev/nvme1 -n1 -c1	sudo nvme create-ns /dev/nvme1 -s 5000000000 -c 5000000000 -b 512 sudo nvme attach-ns /dev/nvme1 -n1 -c1

**The additional namespaces were tested by repeating Steps 1 and 2, but replacing the namespace parameter value so the sectors match the desired namespace capacity[6].*

Test Results

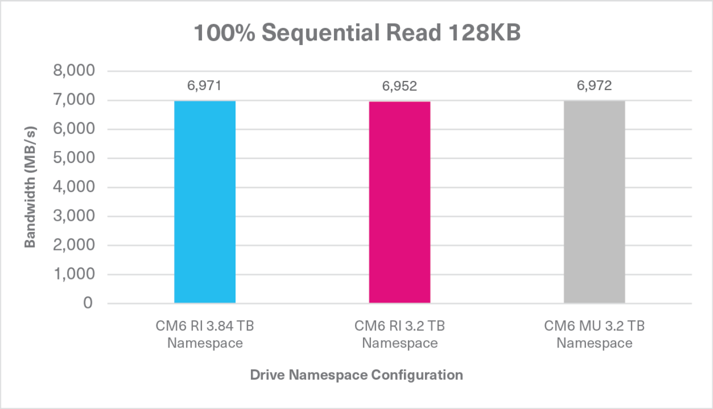
The objective of these seven FIO tests was to demonstrate that a ***1DWPD CM6 Series SSD can perform comparably in write performance to a 3DWPD CM6 Series SSD*** by using NVMe namespaces and reducing capacity. The throughput (in megabytes per second or MB/s) and random performance (in input/output operations per second or IOPS) were recorded.

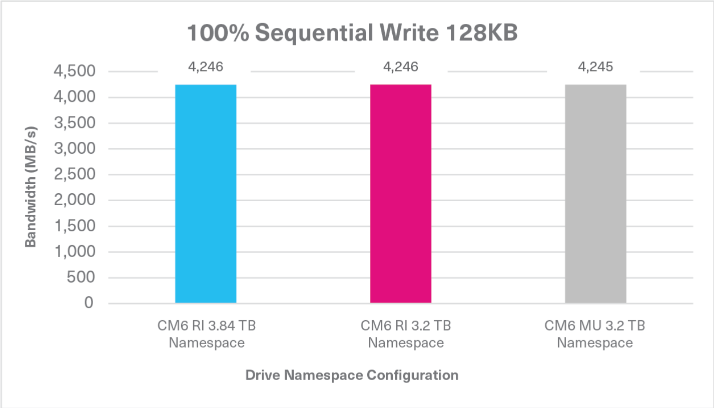
Sequential Read/ Write Operations: *Read and write data of a specific size that is ordered one after the other from a Logical Bus Address (LBA)*
Random Read/ Write / Mixed Operations: *Read and write data of a specific size that is ordered randomly from an LBA*

Snapshot of Results:

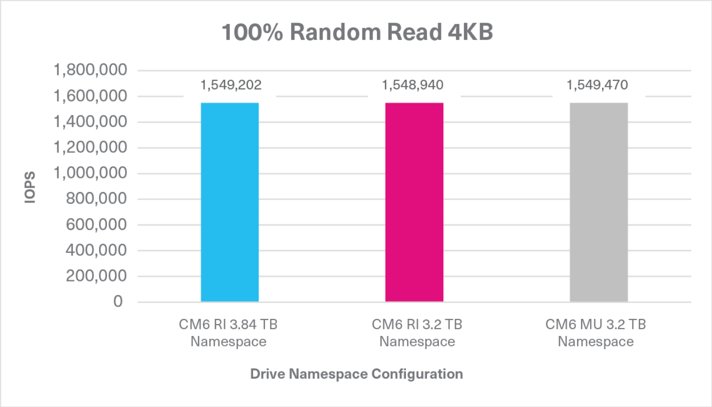
Performance Test	**1st Test Run:** 3.84TB Capacity 3.84TB Namespace Size	**2nd Test Run:** 3.84TB Capacity 3.20TB Namespace Size	**3rd Test Run:** 3.20TB Capacity 3.20TB Namespace Size
100% Sequential Read *Sustained, 128KB, QD16*	6,971 MB/s	6,952 MB/s	6,972 MB/s
100% Sequential Write *Sustained, 128KB, QD16*	4,246 MB/s	4,246 MB/s	4,245 MB/s
100% Random Read *Sustained, 4KB, QD32*	1,549,202 IOPS	1,548,940 IOPS	1,549,470 IOPS
100% Random Write *Sustained, 4KB, QD32*	173,067 IOPS	337,920 IOPS	354,666 IOPS
70%/30% Random Mixed *Sustained, 4KB, QD32*	386,789 IOPS (R) +165,783 IOPS (W) 552,572 IOPS	555,810 IOPS (R) +238,225 IOPS (W) 794,035 IOPS	561,352 IOPS (R) +240,528 IOPS (W) 801,880 IOPS
50%/50% Random Mixed *Sustained, 4KB, QD32*	170,515 IOPS (R) +170,448 IOPS (W) 340,963 IOPS	321,712 IOPS (R) +321,757 IOPS (W) 643,469 IOPS	325,993 IOPS (R) +325,987 IOPS (W) 651,980 IOPS
30%/70% Random Mixed *Sustained, 4KB, QD32*	73,596 IOPS (R) +171,719 IOPS (W) 245,315 IOPS	142,434 IOPS (R) +332,412 IOPS (W) 474,846 IOPS	149,938 IOPS (R) +349,826 IOPS (W) 499,764 IOPS

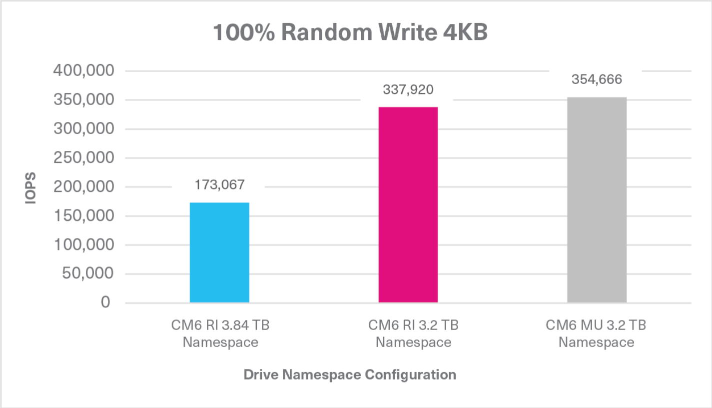
Tests 1 & 2: 100% Sequential Read / Write



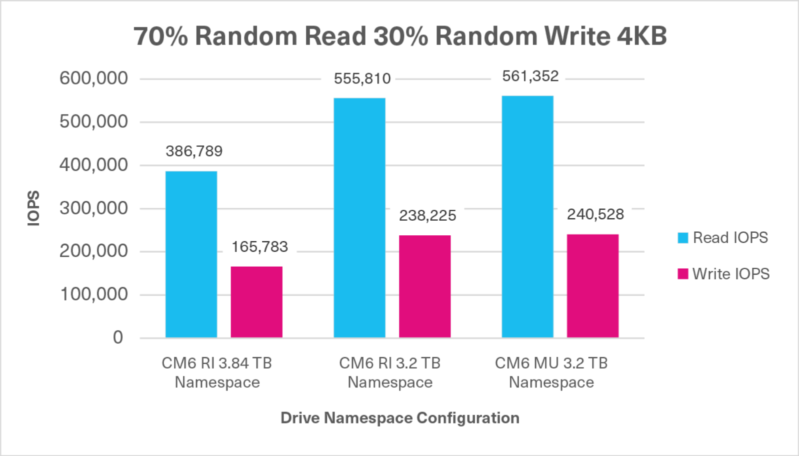


Tests 3 & 4: 100% Random Read / Write

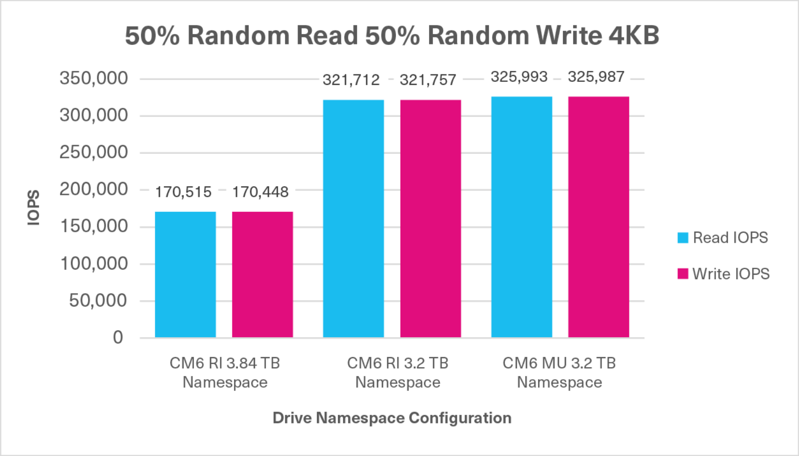




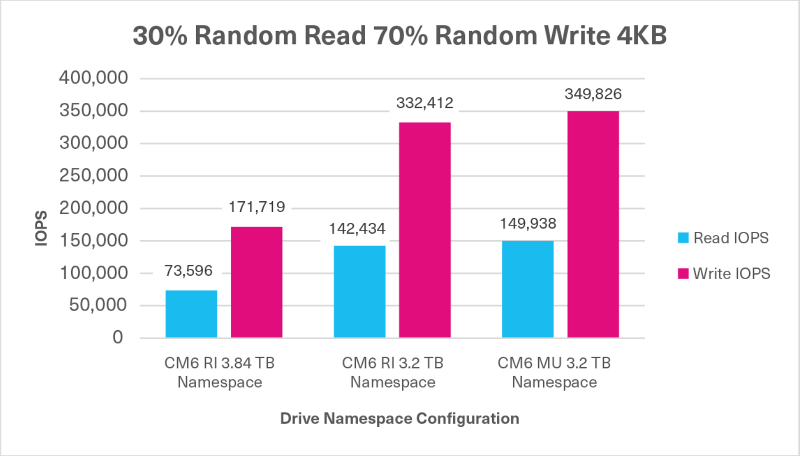
Test 5: Mixed Random - 70% Read / 30% Write



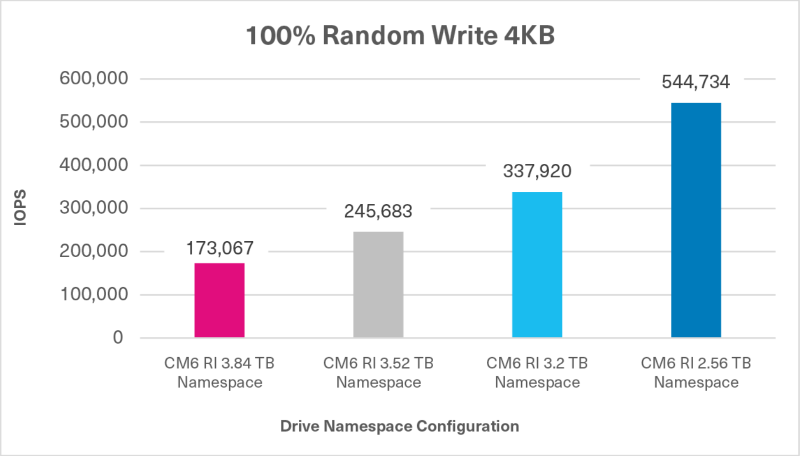
Test 6: Mixed Random - 50% Read / 50% Write



Test 7: Mixed Random - 30% Read / 70% Write



Additional Test: 100% Random Write Using 4 Namespace Sizes



The objective of these 100% random write FIO tests was to demonstrate the increase in random write performance when using NVMe namespaces of different sizes, and reducing capacity. The random performance was recorded in IOPS.

Test Analysis

When a read or write operation is either 100% sequential or random, the performance differences between the three CM6 Series configurations were negligible based on the four FIO tests. However, when the three mixed FIO workloads were tested, the CM6 Series enabled the flash memory that was not provisioned for a namespace to be added back into the OP pool, and demonstrated higher write performance. Therefore, when provisioned with smaller namespaces, in conjunction with reducing the capacity requirements, the 3.84TB capacity, 1DWPD drive performed comparably to a 3.2TB capacity, 3DWPD drive as demonstrated by the test results. Though the 3.84TB capacity / 3.84TB CM6 Series SSD did not perform exactly to the CM6 Series 3.2TB capacity / 3.2TB namespace size SSD, the performance results were very close.

Also evident is a significant increase in the random write performance based on the allocated capacity given to a namespace, with the remaining unallocated capacity going into the OP pool courtesy of KIOXIA firmware. This enables users with finer control over the capacity allocation for each application in conjunction with the write performance required from that presented storage namespace to the application.

ASSESSMENT: If a user requires higher write performance from their CM6 Series PCIe 4.0 enterprise NVMe SSD, using NVMe namespaces can achieve this objective.

Summary

Namespaces can be used to manage NVMe SSDs by setting the random write performance level to the desired requirement, as long as IT administration (or the user) is willing to give up some capacity. With the reality that today's workloads are very mixed, the ability to adjust the random performance means that these mixed and I/O blender effect workloads can get maximum performance simply by giving up already unused capacity. Don't let stranded, unused capacity go to waste when the random performance workload can be maximized!

If longer drive life is the desired objective, then using smaller namespaces to increase the OP pool is a very effective method to manage drives. Enabling these drives to be available for other applications and workloads maximizes the use of the resource as well as its life. However, the use of smaller namespaces to increase drive performance of 100% random write operations and mixed random workloads will show substantial benefit.

Additional CM6 Series SSD information is available here.

Notes:

[1] Definition of capacity - KIOXIA Corporation defines a kilobyte (KB) as 1,000 bytes, a megabyte (MB) as 1,000,000 bytes, a gigabyte (GB) as 1,000,000,000 bytes and a terabyte (TB) as 1,000,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1Gbit = 2^{30} bits = 1,073,741,824 bits, 1GB = 2^{30} bytes = 1,073,741,824 bytes and 1TB = 2^{40} bytes = 1,099,511,627,776 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

[2] Drive Write(s) per Day: One full drive write per day means the drive can be written and re-written to full capacity once a day, every day, for the specified lifetime. Actual results may vary due to system configuration, usage, and other factors.

[3] Flexible I/O (FIO) is a free and open source disk I/O tool used both for benchmark and stress/hardware verification. The software displays a variety of I/O performance results, including complete I/O latencies and percentiles.

[4] Sanitize Instant Erase (SIE) drives are compatible with the Sanitize device feature set, which is the standard prescribed by NVM Express, Inc. It was first introduced in the NVMe v1.3 specification, and improved in the NVMe v1.4 specification, and by the T10 (SAS) and T13 (SATA) committees of the American National Standards Institute (ANSI).

[5] Self-Encrypting Drives (SEDs) encrypt/decrypt data written to and retrieved from them via a password-protected alphanumeric key (continuously encrypting and decrypting data).

[6] To determine the number of sectors required for any size namespace, divide the required namespace size by the logical sector size. Using 2.56 TB as an example, 2.56 TB = 2.56 x 10^12B. Since many SSDs typically have a 512B logical sector size, divide (2.56 x 10^12B) by 512B, which equals 5,000,000,000 sectors.