Redbooks
ibm.com/redbooks

IBM Storage and the NVM Express Revolution

Ioannis Koltsidas
IBM Research Staff, Technical Leader
for Flash and SDS

Vincent Hsu
IBM Systems, IBM Fellow and CTO for Storage and SDE

Storage

In partnership with
IBM **Academy of Technology**

Point-of-View

The need for a standardized and efficient protocol for storage access

Highlights

The industry recognizes the need for standardized and efficient protocols and interfaces that are optimized for low-latency, performance-dense storage media (such as flash) and non-volatile memory technologies. This paper provides the following highlights:

- The NVM Express (NVMe) technology enables storage accesses with low latency, high efficiency, and high scalability.
- The NVMe technology can make server-based SDS higher performing and more efficient.
- IBM Storage has a roadmap that incorporates these technologies in both SDS systems and in integrated appliances that combine software and hardware.
- Use cases provided as examples.

Efficient access to solid state storage is becoming increasingly important as data volumes increase and as data-hungry workloads become more critical to the success of businesses and organizations. With the introduction of IBM® FlashSystem all-flash arrays, IBM pioneered efficient, high-performance access to flash-based storage. This method uses a system architecture that minimizes software interactions in the data path and that builds on efficient interfaces to access purpose-engineered dense flash modules with consistently low latency and high throughput. Purpose-built hardware helps to satisfy the most demanding enterprise workload requirements and to translate raw storage performance into application-level benefits that can unlock new use cases and business value.

With the increasing maturity and wider adoption of flash-based storage in data centers, the broader industry has recognized the need for standardized, efficient protocols and interfaces that are optimized for low-latency, performance-dense storage media (such as flash) and upcoming non-volatile memory technologies.

This IBM Redbooks® Point-of-View publication presents important aspects of the NVM Express (NVMe) protocol and its different flavors as well as the new paradigm for storage access that it enables. It discusses the implications of the NVMe protocol for the enterprise storage industry in general and the transformative effect it has for IBM Storage in particular. The paper presents experimental results that demonstrate the potential of the new technologies.

The NVMe protocol and related technologies

The NVMe protocol is an open collection of standards and interfaces that fully exposes the benefits of non-volatile memory in all types of computing environments, from mobile to data center.[1] It is designed to deliver high bandwidth and low latency storage access. The sections that follow describe the NVMe protocol and interface as it relates particularly to flash-based architectures.

---

[1] `http://www.nvmexpress.org`

Why the NVMe protocol is different

The NVM Express consortium develops the NVM Express specification and describes how NVMe differentiates itself from other interfaces, such as SATA and SAS, as follows:

> *NVM Express (NVMe) is an optimized, high-performance scalable host controller interface designed to address the needs of Enterprise and Client systems that utilize PCI Express-based solid-state storage. Designed to move beyond the dark ages of hard disk drive technology, NVMe is built from the ground up for non-volatile memory (NVM) technologies. NVMe is designed to provide efficient access to storage devices built with non-volatile memory, from today's NAND flash technology to future, higher-performing, persistent memory technologies.*
>
> *There are several performance vectors that NVMe addresses, including bandwidth, IOPs, and latency. For example, the maximum IOPs possible for Serial ATA was 200,000, whereas NVMe devices have already been demonstrated to exceed 1,000,000 IOPs. By supporting PCI Express and Fabrics such as RDMA and Fibre Channel, NVM Express can support much higher bandwidths than SATA or SAS (e.g., a PCI Express Gen3 x4 delivers 4 GB/s). Finally, next generation memory technologies may have read access latency under a microsecond, requiring a streamlined protocol that enables an end-to-end latency of under 10 microseconds, including the software stack.*
>
> *NVMe is a completely new architecture for storage, from the software stack to the hardware devices and systems.*

NVMe over PCIe

The NVMe protocol is an interface specification for communicating with storage devices. In that sense, it is functionally analogous to other protocols, such as SATA and SAS. In contrast, however, the NVMe interface was architected from the ground up for extremely fast storage media, such as flash-based solid-state drives (SSDs) and low-latency non-volatile storage technologies. In particular, NVMe storage devices are typically directly attached to a host system over a PCI Express (PCIe) bus. That is, the NVMe controller is contained in the storage device itself, alleviating the need for an additional I/O controller between the CPU and the storage device. This architecture results in lower latency, throughput scalability, and simpler system designs.

PCIe-attached SSDs are not something new, and devices from multiple vendors have been available in the market for quite a while, each coming with its own proprietary software drivers. However, the NVMe protocol introduces an open and standardized way to access such SSDs. The NVMe protocol enables interoperability for different vendors, architectures, and operating systems, which in turn fosters a wide adoption by server and storage platforms. As a result, NVMe devices become more pervasive, and the increase in volumes drives down the cost.

Importantly, the register-level interface and command protocol that the NVMe interface introduces presents multiple benefits for system-level efficiency and performance. First, by design, the NVMe architecture is lean and simple, with a streamlined command set. All communication is based on a mechanism of paired submission and completion command queues, which use fixed-size entries that can be fetched with a single memory access and processed quickly (as illustrated in Figure 1).

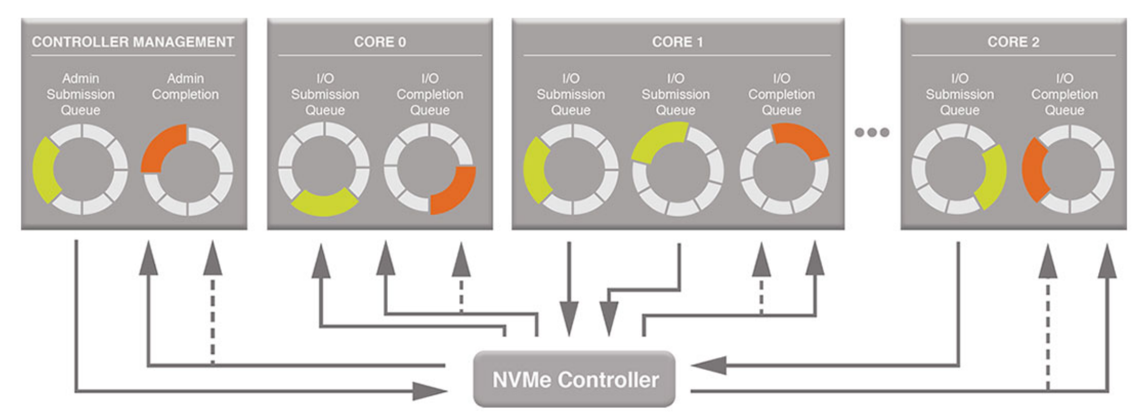

Figure 1 NVMe submission and completion queues[2]

---

[2] Image courtesy of NVM Express organization (`http://www.nvmexpress.org/`)

Work items are queued and de-queued in an asynchronous fashion, requiring no synchronization between the producer and the consumer. Moreover, these queues can be deep (up to 64 K entries per queue), which enables a host to achieve extreme throughput from devices with high degrees of internal parallelism. Just as importantly, the specification allows a large number (up to 64 K) of individual queues. In a multi-core system, each CPU core can allocate its own submission and completion queues independently and can process I/O requests without any synchronization with other cores and without access to shared resources that would entail locking. Furthermore, with appropriate interrupt steering, a completion interrupt can be handled on the CPU core that submitted the respective I/O request.

As a result, each NVMe request can be processed with less software code than traditional protocols, which translates to lower latency and less CPU utilization. At the same time, the distributed lock-free processing of NVMe requests translates into throughput scalability, that is, I/O operations per second (IOPS) that scale with the number of NVMe SSDs in a host.

Beyond these performance benefits, the NVMe protocol offers features that storage systems can take advantage of. It supports end-to-end data protection and data integrity checking, as well as standard security protocols, such as the Opal specification of the Trusted Computing Group. In addition, the NVMe protocol has storage virtualization capabilities in the form of logical volumes, which are called *namespaces*, and is capable of reservation-based, multi-pathed access to namespaces via multiple controllers. For advanced use cases, the protocol offers Dataset Management (DSM) commands that enable the host application to tag logical data with attributes that are related to read and write access frequency, compressibility, sequential or random access, and so on, which the NVMe controller can take advantage of to optimize placement of data in the storage device. Finally, the NVMe interface achieves savings in power consumption via advanced power management that takes advantage of processor power states.

NVMe SSDs are available in multiple form factors, including add-in cards, 2.5-inch SFF drives and M.2 expansion cards, with capacities of multiple TBs per drive. Dual-ported drives are also available to serve the multi-pathed access requirements of enterprise storage. NVMe SSDs are supported on all major server platforms. In addition, NVMe drivers are supported on most operating systems, including open source and commercial. Using such drivers, NVMe SSD namespaces appear as standard-block devices on which file systems and applications can be deployed without any modification.

NVMe over Fabrics

The PCIe bus, however, presents certain challenges for storage systems at scale. Practical challenges limit the number of NVMe drives that can be attached to a host over PCIe to a few tens of devices. Because the maximum length of PCIe cabling is only a few meters, the flexibility of deploying PCIe NVMe drives outside the host server is severely limited and data center level scalability is not feasible. Furthermore, the robustness and error handling capabilities of PCIe are challenged by the requirements of storage systems. For example, surprise hot plug and removal of devices is not supported and might cause host crashes (although such support is expected to be offered in future systems).

NVMe over Fabrics (NVMe-oF) overcomes the limitations of the PCIe bus by extending the benefits of low latency and high efficiency of the NVMe technology across network fabrics to support sharing of NVMe storage at a large scale (100s or 1000s of devices) and over distance. The standard itself is relatively new, because the first version was published in June 2016 and has provisions to support many different network fabric technologies (as illustrated in Figure 2 on page 4). NVMe over Fabrics defines how NVMe commands, responses, and data transfer can be encapsulated over an abstract message-based NVMe Transport layer.

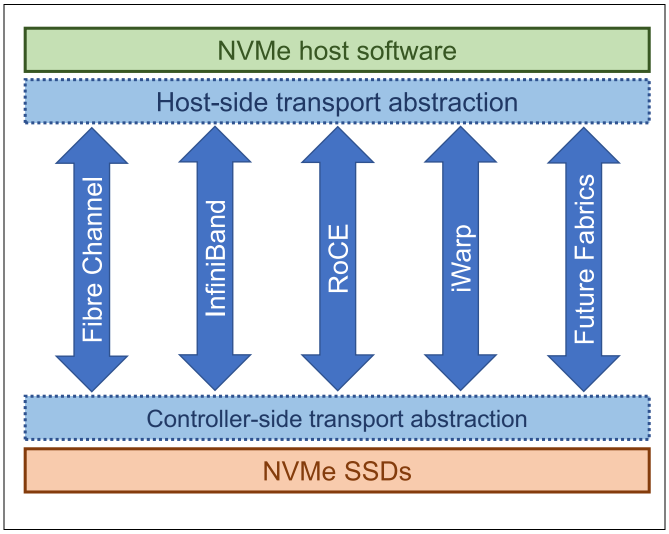

Figure 2 NVMe over Fabrics supports multiple network technologies as transports

The NVMe transport abstraction can be mapped to different network fabric technologies. Currently, the NVMe transport abstraction supports the following main fabric transports:

- NVMe over Fabrics using Fibre Channel is referred to in this paper as *FC-NVMe*.
- NVMe over Fabrics using RDMA is referred to in this paper as simply *NVMe over Fabrics*.

FC-NVMe uses Fibre Channel Protocol (FCP) as its underlying transport, which already puts the data transfer in control of the target and transfers data direct from host memory, similar to RDMA. In addition, FC-NVMe allows for a host to send commands and data together (first burst), eliminating the first data “read” by the target and providing better performance at distances.

NVMe over Fabrics relies on standard RDMA abstractions and, therefore, can be used with different RDMA-enabled networking technologies, including InfiniBand and RDMA-enabled Ethernet. The main RDMA-enabled Ethernet technologies include:

- iWarp, which is layered on top of TCP/IP and, therefore, does not require a lossless Data Center Bridging (DCB) fabric
- RoCE, which is based on InfiniBand transport over Ethernet, and RoCEv2, which supports routability

RoCE requires a lossless DCB fabric, which is a common feature of enterprise network switches. As such, each of these network technologies are viable alternatives as fabrics for NVMe over Fabrics.

The NVMe over Fabrics interface uses the same model of submission and completion queues as PCIe NVMe. As such, it maintains the same asynchronous submission and completion model and achieves similar benefits in terms of latency, efficiency, and scalability as the NVMe technology due to the shortened code paths and lockless concurrency in multi-core environments. Furthermore, the NVMe multi-queue interface maps nicely to the RDMA Queue-Pair model. That is, NVMe queue entries are encapsulated into fabric-neutral capsules, which are in turn transferred using RDMA send/receive queues.

An NVMe over Fabrics initiator on the hosts accesses remote targets (for example, storage endpoints) on the storage node (as shown in Figure 3). For a user on the host, this process is similar to accessing local NVMe SSDs, from both a functional perspective and from a performance perspective. NVMe over Fabrics aims to provide access to a remote NVMe drive with no more than 10 microseconds (µs) of additional latency over a native NVMe drive inside a server. This low-latency access is made possible by using the same commands and structures as PCIe NVMe, which means that they can be transferred end to end, alleviating to a large extent the need for protocol translation between the network protocol and the local interface. Importantly, the simplicity of the technology affords hardware offloading for much, if not all, of the functions performed by the NVMe over Fabrics target.

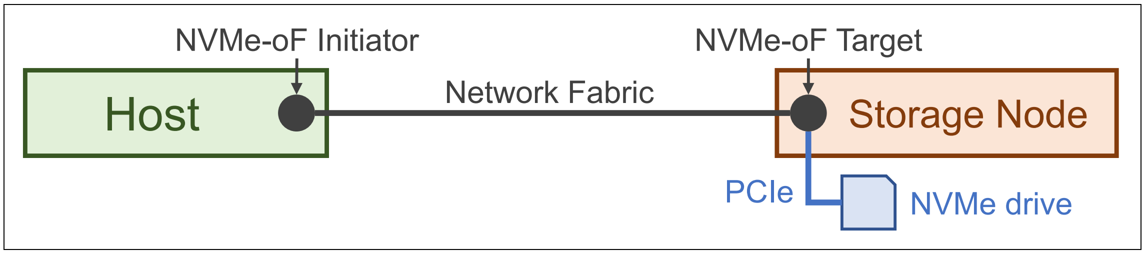

Figure 3 Simple NVMe over Fabrics topology

NVMe over Fabrics is not the first networked storage protocol to take advantage of RDMA networking. Existing protocols, such as the iSCSI Extensions for RDMA (iSER) and the SCSI RDMA Protocol (SRP), use RDMA to bypass the host networking software stack and are shown to deliver performance improvement compared to traditional protocols. Although at the network transport layer these protocols are as efficient as NVMe over Fabrics, the latter introduces the following advantages:

- NVMe over Fabrics supports in-band management functions, including the capability to create, manage, and destroy namespaces, which is not supported by SCSI-based protocols.
- When exporting NVMe controllers directly over the network, NVMe over Fabrics can avoid protocol translation, because I/O command and response structures can be transferred unchanged end to end.
- The host stack benefits from shorter code paths, because it completely avoids the host SCSI layer.
- NVMe over Fabrics simplifies systems and applications that access both local and remote storage resources, because it enables access to both using mostly the same interface.

Driver support: The required kernel drivers for NVMe over Fabrics initiator and target function are already implemented in the upstream Linux kernel, and it is expected that they will ship with enterprise Linux distributions in the future. Likewise, other platforms and operating systems are expected to implement the necessary support for the protocol.

Being a relatively new technology, however, NVMe over Fabrics needs to evolve to reach the level of maturity required by enterprise storage applications. Improvements must continue to address certain aspects of graceful error handling and recovery from errors that are associated with the network transport needed to allow enterprise storage systems to achieve the required levels of fault tolerance and high availability. In addition, work must continue in order to achieve the required levels of scalability in enterprise environments with tens of thousands of exported namespaces and thousands of hosts accessing them. Furthermore, to best serve enterprise virtualized environments, it is expected that over time the standard will enable virtual machines (VMs) and containers to offload operations to the storage layer, as has happened with SCSI-based protocols, and will enable the storage system to offload data services functions (for example, copy services) to the NVMe layer. In addition, improvements in the traditional SCSI stack to remove locking and serialization overheads are expected, which might potentially narrow the advantage that NVMe over Fabrics has today and might help provide more of the benefit of RDMA-based protocols to traditional environments.

User-space I/O architectures

As discussed previously, standard NVMe and NVMe over Fabrics drivers expose NVMe storage as standard block devices, which can be used by unmodified file systems and applications. However, the simplicity of the new protocols enables a new paradigm of accessing storage, one in which the application runs the storage driver in the user-space, foregoing the block device abstraction altogether. In this new paradigm, the software driver is linked directly into the application code, and storage is accessed by manipulating the memory-mapped submission and vy completing NVMe queues directly from the user-space, completely bypassing the operating system kernel in the data path.

The control for storage accesses moves up in the stack, from the operating system kernel to the application runtime. The application can now exercise *poll-based I/O completions*, as opposed to *interrupt-based completions*. In addition, by carefully distributing I/O request processing to CPU cores in alignment with the application threads, it can exercise I/O in a lockless and cache-friendly way. By eliminating kernel and interrupt context switching, a great deal of scheduling overhead and data copying is avoided. The code path per I/O request is shortened, too. As a result, the performance and efficiency gains of NVMe and NVMe over Fabrics are increased. As shown in the examples that follow, millions of I/O operations can be served using a single CPU core.

Open-source implementations of the new paradigm have appeared in the community, including the Storage Performance Development Kit (SPDK) by Intel[3] and UNVMe by Micron[4]. Importantly, the simplicity of NVMe and NVMe over Fabrics and the commonality between the two interfaces make it practical and relatively straightforward to develop a user-space driver that is tailored to the specific needs of a data-intensive application.

As is evident from Figure 4, bypassing the kernel also means giving up functions and services that the kernel provides. The responsibility for implementing such functions now falls to the application. For example, the application now needs to manage itself: virtualization, durability of writes, recovery from crashes, data organization in a device, a global namespace across devices, access to a device over multiple paths, and so on. Therefore, it is expected that applications that adopt the new paradigm will be applications that have some combination of the following characteristics:

- Extremely data-intensive
- Will benefit significantly from low-latency
- Have other reasons to implement kernel functions on their own
- Don’t need such functions at all

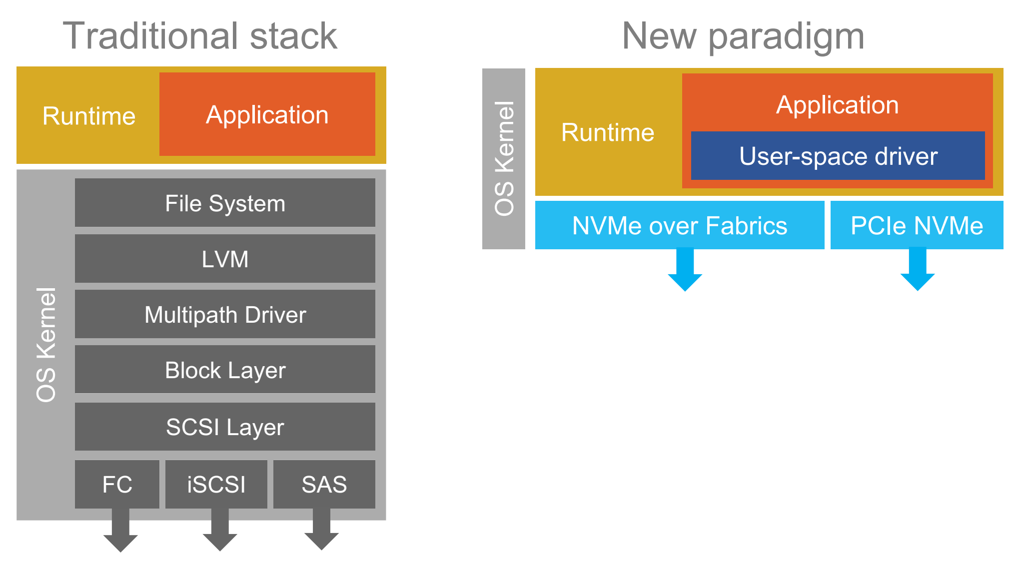

Figure 4 The traditional I/O stack and the new paradigm

---

[3] `http://www.spdk.io`
[4] `https://github.com/MicronSSD/unvme/`

Micro-benchmarks

We now turn to an experimental evaluation of the protocols using micro-benchmarks. These benchmarks used an x86-based server that was equipped with 2 x Intel Xeon E5-2630 v4 CPU nodes and 128 GB of memory, to which we attached 12 Intel P3600 Flash-based NVMe SSDs over the PCIe bus (6 SSDs on each CPU node). The benchmarks used a flexible I/O tester synthetic benchmark (also known as *fio*) as the workload generator for the experiments and used the block device interface exposed through the kernel. In addition, they used a custom I/O workload generator based on SPDK for the user-space I/O experiments.

The first benchmark measured the total system throughput in terms of IOPS for a 4 KiB random read workload. It increased the request rate (queue depth) to reach the maximum throughput. Figure 5 shows the measurements, both for a single SSD and the aggregate for 12 SSDs. As shown in the results, this benchmark achieved an extremely high throughput. Using the user-space I/O architecture with SPDK completely saturated all 12 drives, reaching a total throughput of 3.9m IOPS.

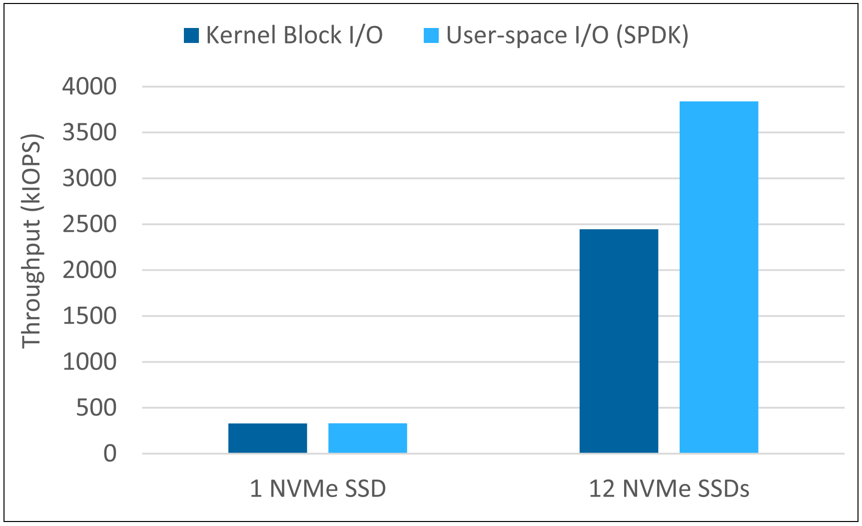

Figure 5 Throughput for 4 KiB random reads

The next benchmark evaluated the efficiency of the NVMe protocol in terms of IOPS per CPU core. To that end, this test disabled all CPU cores but one in the system and then used that core to run both the OS and the workload generator. Figure 6 on page 8 illustrates the throughput measured when using a local SATA SSD through the kernel interface, an NVMe SSD through the kernel interface, and the same NVMe SSD using the user-space I/O paradigm (with SPDK). In all three cases, the single CPU core was the bottleneck in performance, that is the maximum throughput that one core could serve was measured. As demonstrated by these results, NVMe is more efficient than SATA in the kernel implementation, achieving 76% more IOPS per core. However, the most dramatic improvement in efficiency comes when the NVMe driver is moved to the user-space, where a 9x improvement is achieved compared to accessing the NVMe driver over the kernel block device.

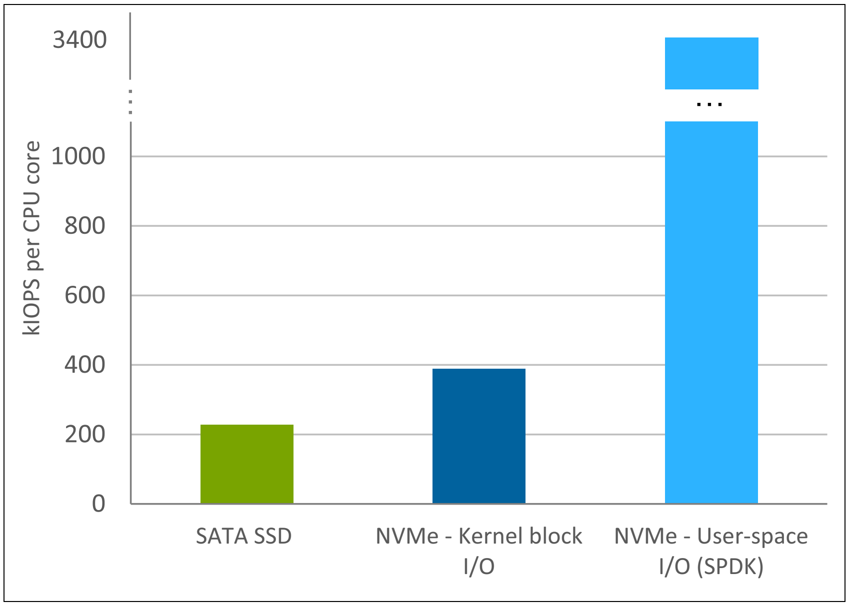

Figure 6 IOPS throughput using a single CPU core

The next experiment measured the latency when accessing NVMe devices. In addition to the Flash-based NVMe SSDs used previously, this benchmark also experimented with Intel Optane P4800 NVMe SSDs, which use the 3D-XPoint memory technology instead of the Flash. Figure 7 illustrates the median latency measured for 4 KiB random read requests at a queue depth of 1 (that is, when only a single request is outstanding at the device). For this test, the kernel-based access to the NVMe devices imposed a latency overhead of 8 microseconds. Thus, for the Flash-based SSD, only a 10% improvement in end-to-end latency occurred. For the 3D-XPoint device, however, the latency benefit of the user-space access translated to 2.3X lower end-to-end latency.

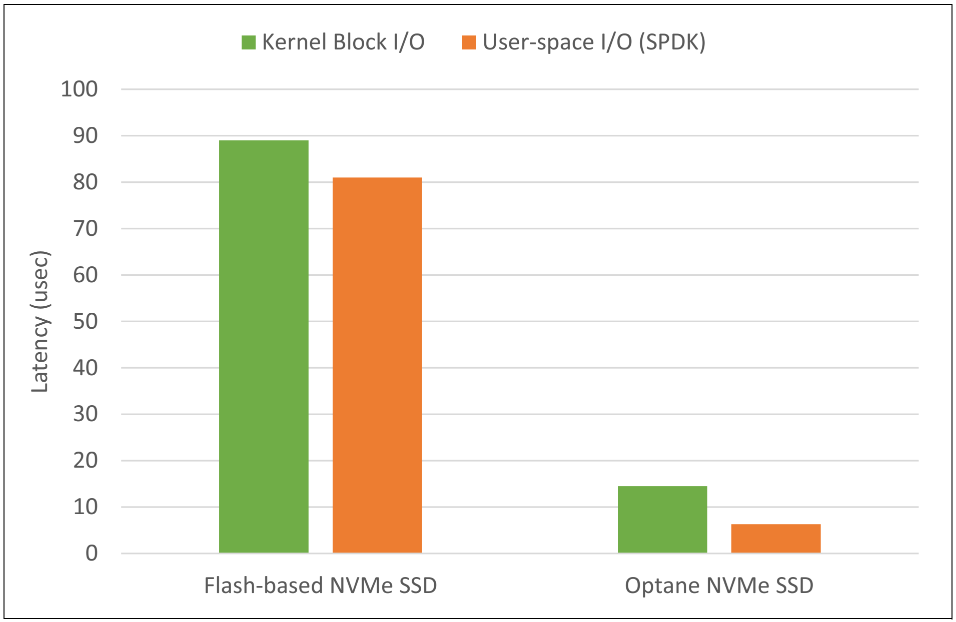

Figure 7 Median latency for 4 KiB random reads

Overall, the micro-benchmarks presented in this section confirmed expectations. NVMe provides high IOPS density and low latency. With a user-space I/O architecture, NVMe offers dramatic improvements in terms of IOPS/core efficiency. For devices built out of next-generation memory technologies, such as 3D-Xpoint, the user-space I/O architecture translates into substantial latency savings as well.

IBM and the NVMe technology

IBM has a long history of commitment to industry standards, and the NVM Express organization is no exception. As a member of the NVMe workgroup, IBM aims to contribute to the standards and to lead the enablement of enterprise storage systems that use the new drives, network protocols, and I/O architectures. We expect that the NVMe protocol and related technologies will permeate multiple levels of the storage hierarchy and serve multiple use cases. As such, it will have a broad impact for the entire IBM storage portfolio.

IBM's roadmap incorporates the new technologies in both software-defined storage systems and in integrated storage systems that combine software and hardware. To accommodate the requirements of a wide range of applications and client environments, we provide offerings that are tailored to workloads of different types and sizes. IBM will adopt flash-based NVMe drives and drives that are based on 3D-Xpoint memory for systems where extremely low latency is critical. Importantly, we also plan to optimize IBM hardware and software end to end to deliver the potential of the NVMe protocol efficiently all the way to client applications. Our plan includes NVMe over Fabrics, which is used both for host attachment and in the backend fabric of IBM systems. We expect that different flavors of NVMe over Fabrics might be more suitable for different environments, including NVMe over InfiniBand, iWarp Ethernet, RoCE Ethernet, and Fibre Channel. In addition, we plan to support the different NVMe over Fabrics incarnations appropriately in the IBM portfolio.

As the IBM roadmap is implemented, we plan to work with IBM clients and IBM Business Partners to identify how the technologies best benefit their use cases and workloads. We can help them transition to an NVMe infrastructure. In addition, IBM plans to work with the entire ecosystem and the wider industry, including storage, server, and networking platforms to ensure a smooth and effective transition.

Software-defined storage

One advantage of NVMe interface is that it shortens code paths that are required during I/O processing and also enables bypassing hypervisors and operating systems, including traditional storage drivers. Effectively, it enables IOPS scalability in a CPU-efficient way. As more NVMe drives are attached to a server, software can take advantage of the greater throughput potential with high CPU efficiency. Therefore, NVMe can enable server-based software defined storage solutions to deliver higher performance with greater efficiency. It is expected that NVMe and NVMe over Fabrics will first impact server-based SDS (as compared to other types of enterprise storage).

The NVMe interface enables the following potential use cases:

- Use NVMe drives as a caching medium, either on the host or in the storage system, to reduce the read latency for cache-friendly workloads. The local read-only cache (LROC) function of IBM Spectrum™ Scale is already capable of using NVMe drives in this way. Recent experiments demonstrated that IBM Spectrum Scale™ using NVMe flash drives increase throughput by 25%, while at the same time improving the overall response times significantly in the SPEC SFS 2014 benchmark.[5]
- Use NVMe storage as the fastest-possible storage tier in a hybrid storage system. In certain cases, it is expected that this storage tier can extend to the storage clients, potentially trading off fault tolerance and high availability guarantees in favor of performance. Such a use case might be suitable for data pipelines that produce high volumes of temporary, performance-critical data.
- Provide an all-NVMe system that is optimized end to end for performance. This use case entails changing the entire stack to make it well suited for NVMe and using the user-space I/O paradigm to gain performance, efficiency, and IOPS density. Over time, it is expected that additional hardware assists will be put in place to accelerate NVMe functions associated with data services beyond the I/O itself, such as compression. IBM Spectrum Virtualize™ and IBM Spectrum Accelerate™ will follow this approach.
- Take advantage of NVMe over Fabrics as a protocol for a host attachment. This method will complement existing RDMA-based protocols, such as iSER and SRP, which offer similar performance to NVMe over Fabrics. It is expected that the existing protocols will play an important role until NVMe over Fabrics reaches maturity and beyond. NVMe over Fabrics will have additional advantages, such as in-band management, shorter code

---

[5] `https://www.spec.org/sfs2014/`

paths on the host, and a common interface with local NVMe. The latter advantage is important for applications that are optimized for user-space NVMe access and that want to maintain a similar storage access driver for both local and remote accesses. Therefore, it is expected in the near-term that NVMe over Fabrics will focus on specific optimized applications. In the medium-term, the traditional SCSI stack and NVMe over Fabrics capabilities converge, and this will be the point when traditional SCSI-based environments can migrate to receive the full benefits of NVMe over Fabrics. IBM Spectrum Virtualize follows this approach, starting with existing RDMA-based protocols (iSER) and transitioning to NVMe over Fabrics as the capabilities of the two technologies converge.

- Use NVMe over Fabrics as a protocol for back-end storage access. A similar approach is already used today by IBM Spectrum Accelerate, which uses the user-space I/O paradigm to access back-end storage over RDMA in the FlashSystem A9000 and A9000R. IBM Spectrum Accelerate is planned to extend support for this type of back-end access with NVMe over Fabrics.

Integrated storage systems

In the IBM storage portfolio, integrated storage systems for enterprise block storage, such as IBM Storwize® family, FlashSystem V9000 and A9000, are built with IBM Spectrum Virtualize and IBM Spectrum Accelerate software-defined storage. This integration means that the innovations and across-the-stack optimizations for NVMe and NVMe over Fabrics technologies that are developed in the IBM Spectrum Storage™ portfolio naturally flow into the IBM storage systems, including Storwize V7000, FlashSystem V9000, FlashSystem A9000, and FlashSystem A9000R systems. Similarly, IBM Elastic Storage™ Server file and object systems are built with IBM Spectrum Scale software and will benefit in the same way. IBM storage systems are planned to use NVMe flash and, potentially, NVMe 3D-Xpoint, as well as NVMe over Fabrics storage protocols to reduce latency, increase throughput, and achieve higher efficiency.

Some purpose-built storage systems might still use proprietary protocols and interfaces in the backend because, in certain special cases, they fit the requirements of a specific product better. In the front end, however, IBM products are planned to adopt NVMe over Fabrics in multiple flavors to offer the lowest possible latency and the most CPU-efficient access from the server to storage. This support is particularly important for systems that are engineered specifically for performance, such as IBM FlashSystem® 900, where NVMe over Fabrics is expected to reduce the latency for hosts accessing storage and boost CPU efficiency, which complements the hardware-only data path in the all-flash array itself. Therefore, NVMe over Fabrics attachment is a high priority for FlashSystem 900. NVMe over Fabrics is planned to be delivered over RDMA-enabled Ethernet, either RoCEv2 or iWarp.

IBM FlashSystem is planned to take advantage of newer generations of memory technologies as they become available in the market. IBM plans to continue to apply industry-leading IBM FlashCore® intellectual property to current and future memory technologies, including ever increasingly dense NAND flash and low latency, high endurance 3D-Xpoint. IBM believes that low-latency Tier-0 shared storage will play a critical role for cognitive and analytics workloads. IBM leads this segment with FlashSystem and is well positioned to maintain that leadership in the future by using the next-generation of protocols and memory technologies.

What’s next: How IBM can help

The NVMe interface enables storage accesses with low latency, high efficiency, and high scalability. NVMe over Fabrics (the extension of NVMe for network fabrics) extends that capability across the network by enabling access to remote storage resources over the same interface, with almost the same performance as local, direct-attached storage. The new interfaces and protocols introduce exciting opportunities for software-defined and integrated storage systems, which can now take advantage of NVMe devices based on flash and future non-volatile memories to deliver extremely short response times to applications in a cost efficient and scalable fashion. IBM plans to support the new interfaces and protocols and lead in their adoption in enterprise storage systems. IBM believes that NVMe and NVMe over Fabrics technologies will have a transformational effect to the entire storage industry.

Stay tuned for more updates and product announcements. Contact your IBM representative to find out more about what NVMe-optimized IBM Storage can do for your current and future workloads and applications.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION 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. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

- IBM®
- IBM Elastic Storage™
- IBM FlashCore®
- IBM FlashSystem®
- IBM Spectrum™
- IBM Spectrum Accelerate™
- IBM Spectrum Scale™
- IBM Spectrum Storage™
- IBM Spectrum Virtualize™
- Redbooks®
- Redbooks (logo) ®
- Storwize®

The following terms are trademarks of other companies:

Intel, Intel Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

REDP-5437-00

ISBN 0738456055

Printed in U.S.A.