PAVILION

WHITE PAPER

NVMe-oF Architecture

Taking Data Storage from Evolution to Revolution

Table of Contents

Introduction

Like any technology, enterprise data storage has evolved over time. Server hard drives were replaced by RAID arrays. DAS storage moved to SAN and NAS, only to go back to DAS in rack scale implementations that use software defined storage to aggregate, provision, and manage storage. While server data storage has continued to evolve over the years, compute and networking technologies have seen not just evolutionary, but revolutionary changes.

Processors have gone from early 4-bit processors to multicore to clustered, parallel processing systems. Networking has transformed from a single bus that could connect a few users to the switched environment that powers the modern world. Still, while compute and networking technologies have been transformed, storage technologies have not taken that final leap to truly unlock the power of data.

A Brief History of Storage Interfaces

Server data has historically been stored on hard disk drives (HDDs) which use rotating platters to magnetically store data. As the disks rotate, an actuator arm moves back and forth across the platters to read and write data. Over time disks would spin faster and faster in an effort to reduce the amount of time it took to get the right spot on the disk under the read/write heads on that arm, finally reaching a top speed of 15K RPM. Storage densities have continued to increase over time, but that rotational speed has long since hit the theoretical limit..

HDDs were commonly connected to servers using one of two types of interface, SCSI or ATA. SCSI and ATA each enjoyed a long life, with successive generations increasing the available bandwidth.

ATA offered lower performance than SCSI, was limited in the number of devices that could be supported, and could only connect devices internally to the server. SCSI allowed the connection of a greater number of devices, as well as connectivity to external devices. ATA was only able to connect two devices per channel, with a maximum of two channels per system. SCSI could connect up to 15 devices per channel, with the only limit to the number of channels per system being the number of available bus slots on the server board. The SCSI protocol supported multiple device types, including tape drives, optical drives, scanners, and more. ATA could only support hard drives and, through the ATAPI interface, optical devices such as CD and DVD drives.

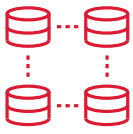
For all it's advantages, SCSI costs significantly more than ATA. As a result, SCSI became the standard for enterprise servers while ATA became the dominant consumer platform. To support the cost differential, device manufacturers would often harden SCSI devices so that they would have a longer mean time between failure (MTBF) than ATA drives.

Both SCSI and ATA used a parallel bus and over time it became harder to maintain data integrity at ever increasing speeds. Eventually, a new solution was needed. ATA and SCSI went from being a parallel bus to a serial bus. SATA (serial ATA) and SAS (Serial Attached SCSI) were born. Each of these interfaces offered greater device connectivity along with higher overall aggregate performance.

DAS, SAN, and NAS

DAS

Direct Attached Storage (DAS) is exactly what it sounds like, Storage devices are directly attached to the system bus through an interface card known as a host bus adapter (HBA). Historically, DAS has offered the highest possible performance in terms of available bandwidth. Some of the challenges with DAS included that data cannot be easily shared across servers, so if something happens to a server, data access will be lost even if the drives are fully functional. Also, at scale, DAS was difficult to manage.



SAN

Storage Area Networks (SAN) were the solution to the DAS problem. Network fabric protocols were used to connect external storage arrays to multiple servers. This allowed for a greater number of devices to be connected to multiple servers simultaneously. Data could now be shared.

NAS

SANs enabled multiple servers to connect to block level storage to share data and increase availability. SANs changed how data was stored in the enterprise. Network Attached Storage (NAS) did the same for file level data. Yet for all the benefits of SAN and NAS storage, they could never match the performance of DAS.

Rack scale systems utilize DAS storage for the best performance and use software to combine and share resources across all systems. While this does provide aggregate performance benefits, running management software on each node adds additional processing overhead, so no system is able to truly deliver the performance it is capable of. The result is that more resources are needed, reducing efficiency and increasing costs.

Flash Memory

Hard disk drives continued to evolve and get faster and denser. While density continues to grow, they have long since reached a maximum practical rotational limit of 15K RPM. One of the biggest drawbacks of the growing density of disk drives is the performance impact to the loss of a drive in a RAID array.

When a drive would fail, parity would be used to identify the data on the lost drive. Each parity checksum would have to be calculated against the remaining data to rebuild the lost data. As the number of drives in arrays grew, and as those drives got bigger, the performance impact of those parity calculations could have a disastrous impact on performance. For large arrays, rebuilding from a failed drive could take hours, or in some cases, days to complete. The more drives in an array, and the larger those drives were, the longer the rebuild would take. If another drive were to fail before the rebuild was complete, then all the data on the entire array would be lost. RAID 6 addressed this by providing the ability to survive two drive failures, but at the cost of even longer rebuild times and lower overall array capacity.

A new solution was needed. The emergence of solid state drives (SSDs) seemed to solve all of the problems. Flash was an order of magnitude faster than traditional hard drives and as flash drives have no moving parts, they are much less prone to failure.

Initially, SSDs used the traditional SAS and SATA interfaces to connect to servers. For all the benefits of flash memory, SSDs were considerably more expensive than hard disk drives and were initially used as caching devices to improve the performance of legacy disk arrays. Using existing interfaces allowed SSDs to be easily connected to existing infrastructure.

As the price of flash went down over time, hybrid disk arrays, which combined SSDs for performance and HDDs from capacity, became common. Still, over time the cost of flash continued to fall and capacity increased. Plus, flash also enabled the use of data compression, which reduced the footprint of data on the flash, further reducing the total capacity needed. Eventually this made the cost of storing data on SSDs comparable to that of using HDDs. All Flash Arrays (AFAs) began to replace disk based arrays in the data center.

Enter NVMe

For all the benefits of flash, the use of SAS and SATA interfaces acted as a bottleneck to SSD performance. Designed for the serial access of HDDs, these interfaces were simply not capable of taking advantage of the parallel access that SSDs were capable of. A new solution was needed.

Non-Volatile Memory express (NVMe) was developed specifically to take advantage of the performance of flash media. Unlike the SAS or SATA protocols, both of which add significant overhead increasing latency, NVMe adds very little latency. NVMe supports 16Gb/s of throughput per device compared to 6Gb/s SATA or 12Gb/s SAS.

NVMe over Fabrics (NVMe-oF) enables the connection of hosts to storage using the NVMe protocol across a fabric. NVMe-oF extends the NVMe block protocol for use over a variety of network fabrics.

A next-generation interface on a legacy architecture

The need to move away from SAS and SATA to take advantage of the performance of flash memory was obvious. The advent of NVMe has changed what is possible.

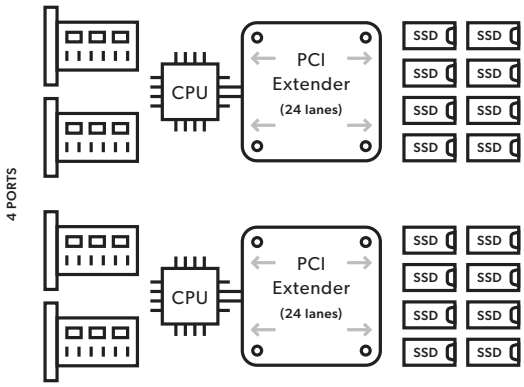
Still, using NVMe SSDs in a legacy storage architecture does nothing but move the bottleneck. AFA storage vendors all offer arrays with NVMe SSDs, yet these arrays are based on a legacy design that prevents them from taking advantage of what NVMe can offer.

There are two problems with legacy array design. The first is that many are still offered with Fibre Channel interfaces. While this does allow organizations to add the arrays to existing SANs, the FC interface hampers AFA performance in the same way that SAS and SATA did. Using NVMe SSDs in a FC array obviates most of the performance benefits that NVMe offers.

The other, and perhaps larger problem, is the fundamental design of these arrays. Since the early days of external SCSI arrays, a dual controller design has been used. The idea was that by using two controllers in an external storage array, high-availability could be achieved. This was because by using two controllers, in the event of loss of access to a controller, a second redundant path could offer continuous access to data through the other controller.

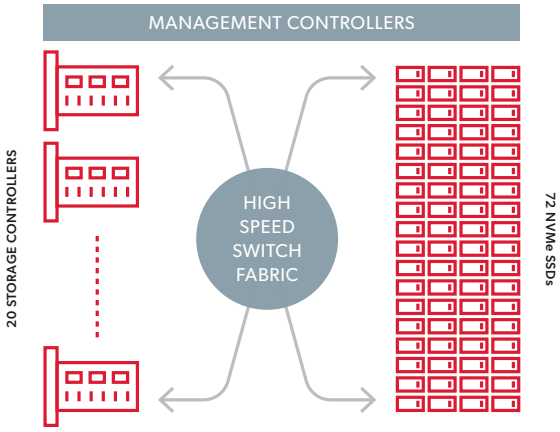
In this design, each controller is a small, dedicated server that has a processor on it that manages the flow of data in a serial fashion. Once again, NVMe, which can enable high performance parallel access to flash, is limited by the serial access of the controller. Even in arrays with active-active controllers, that provide two paths to data, this design dramatically limits the performance of the SSDs.

Typical AFA Single or Dual Storage Controller



- ✕ **Limited expandability:** network & few storage controllers
- ✕ **Limited performance:** number of SSDs, weak CPU
- ✕ Multiple SPOFs
- ✕ Backward compatibility limits performance

Pavilion HFA



- ✓ **Flexibility:** Multiple networking ports and storage controllers
- ✓ **High-performance:** 120 GB/s, 20M IOPS, and 40µs latency
- ✓ No SPOF
- ✓ End-to-end NVMe, no need to support legacy technology

By moving from a legacy, dual controller architecture, overall array performance can be greatly enhanced.

	Traditional Dual Controller AFA	Modern Hyperparallel NVMe Array
Performance	10GB/s at 1M IOPS	>100GB/s with >10M IOPS
Latency	1ms	100µs
SPOFs	Multiple	None
End-to-end NVMe	No	Yes

Eliminating the Array Rebuild Penalty

The parallel access that SSDs offer can provide an additional benefit beyond read/write performance. Earlier we discussed how RAID array rebuild times were impacted as drives became larger and/or more drives were added to the array.

Unlike HDDs which rebuild data in a serial fashion, with the parallel access of an SSD in a RAID array, multiple controllers can work to rebuild different data stripes simultaneously. This enables significantly faster array rebuilds, potentially reducing those rebuild times from days to only minutes. Combined with the longer MTBF for solid state devices, data availability can be greatly increased.

Like replacing a hub with a switch

Unlike legacy, dual controller arrays, a Hyperparallel Flash Array (HFA) is based on the same principle as a network switch. In the early days of ethernet, network hubs only allowed one device to talk on the network at a time. These were eventually replaced by switches, which allowed multiple simultaneous (and faster) connections.

Enterprise networking, the evolution of the Internet, and the modern world we live in would not have been possible without the move from that legacy architecture to a modern switched design. In the same way, Hyperparallel Flash Arrays with NVMe-oF have the same revolutionary change on the way data is accessed and used. With a switch-like design, a Hyperparallel Flash Array can support multiple concurrent data paths, and even different protocols such as block, file, and object simultaneously.

RDMA - Changing the Game

Remote Direct Memory Access (RDMA) is also one of the technologies that is making the move to a switched storage architecture so compelling. Simply put, RDMA enables the transfer of data from the memory of one system to another remote system, without the involvement of the host CPU or OS.

In a traditional DAS system, when an application performs a read or write, the host CPU would handle all the processing of moving the data into and out of system memory. While the CPU was performing those functions, it would take away cycles from other tasks, such as application processing. Early SCSI controllers solved this by using a Direct Memory Access (DMA) engine to manage the movement of data into and out of system memory without involving the host CPU. A DMA engine was simply a second processor on the HBA designed specifically to perform that function. The DMA engine freed the host CPU to perform other functions, dramatically increasing overall performance.

RDMA provides this same benefit using remotely connected devices, so a server that is connected to storage over a fabric can have data moved directly into system memory without interrupting the CPU. Further, RDMA can perform a Zero Copy, which means that in addition to not interrupting the CPU, the network stack is also bypassed, further reducing overhead and driving down latency.

RDMA can be used with either Infiniband or, through RDMA over Converged Ethernet (RoCE), using traditional switched ethernet networks.

Conclusion

Any modern storage environment will require the use of an NVMe architecture that is designed to finally unlock the performance capabilities of flash. But simply using NVMe within a legacy architecture is not enough. To truly take advantage of the benefits that flash storage promises, then the entire infrastructure must be designed for flash.

Using a design that was designed for hard disk drives, such as the legacy dual controller architecture, will bottleneck even the fastest SSDs. In the same way that switched ethernet changed what a could be done on a network, only a hyperparallel storage model can fundamentally reshape what flash storage can offer.

The combination of NVMe and NVMe-oF finally removes all of the legacy storage architecture bottlenecks that have plagued flash and prevented organizations from realizing the benefits that this technology offers. Plus, by using RDMA in conjunction with NVMe-oF, storage can finally be disaggregated from servers, to provide all the benefits of shared storage networking while still delivering the performance of DAS.

> *For more information about NVMe-oF or the Pavilion HFA, visit our website or contact us directly.*
> *Website: http://www.pavilion.io/*
> *Email: info@pavilion.io*
> *Phone: (669) 263-6900*