It's Time to Talk About HPC Storage: Perspectives on the Past and Future

Bradley Settlemyer[1], George Amvrosiadis[2], Philip Carns[3], and Robert Ross[3]

[1]Los Alamos National Laboratory
[2]Carnegie Mellon University
[3]Argonne National Laboratory

September 30, 2021

Abstract

High-performance computing (HPC) storage systems are a key component of the success of HPC to date. Recently, we have seen major developments in storage-related technologies, as well as changes to how HPC platforms are used, especially in relation to artificial intelligence and experimental data analysis workloads. These developments merit a revisit of HPC storage system architectural designs. In this paper we discuss the drivers, identify key challenges to status quo posed by these developments, and discuss directions future research might take to unlock the potential of new technologies for the breadth of HPC applications.

High-performance computing (HPC) storage systems have become trusted repositories for hundreds of petabytes of data with aggregate throughput rates in the terabytes per second. Numerous research advances have contributed to this success. Object storage technologies helped eliminate bottlenecks related to the management of space on storage devices. The development of separate data and metadata planes facilitated scale-out in the data plane to enable high throughput. The adoption of network portability layers eased porting to new HPC networking technologies. Disaggregation was adopted early, bringing powerful cost and administrative savings and providing flexibility to serve the diverse batch workloads typical of HPC. Together with input/output (I/O) middleware technologies, HPC storage systems have largely addressed the throughput challenges of checkpoint and restart for traditional Message Passing Interface (MPI) simulation codes, which was their primary driver for many years.

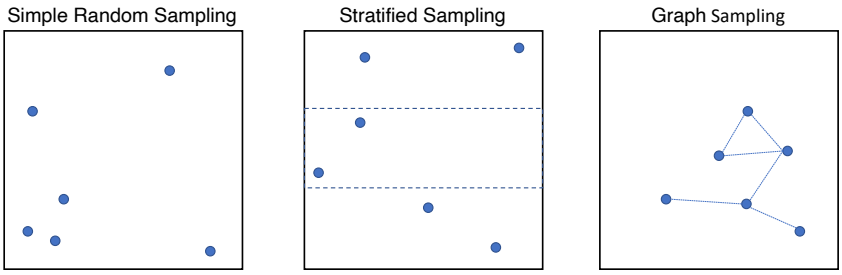

Figure 1: Examples of prevalent analysis access patterns: Data-intensive analysis algorithms (propelled by breakthroughs in AI and statistical methods) must extract samples from immense data sets, thereby triggering storage access patterns that *are unpredictable to outside observers.* These workloads put pressure on the storage system's random read input/output operations per second (IOPS) rate and response time in ways that cannot be solved with general purpose caching and prefetching.

Meanwhile, HPC applications have evolved from numerical simulations to workloads that include Artificial Intelligence (AI) and analytics. For example, scientists at the Oak Ridge National Laboratory (ORNL) Health Data Sciences Institute are developing AI-based natural language processing tools to extract information from textual pathology reports using Summit, the USA's most powerful supercomputer, due to the vast amounts of memory it provides to its compute cores. Similarly, the High Luminosity Large Hadron Collider (HL-LHC) will further extend the capabilities of the LHC, allowing further investigation of phenomena fundamental to the nature of the universe. To be installed in 2025, these enhancements will lead to annual data generation rates of tens of petabytes, with reduced datasets in the petabyte range being used for analysis. These applications are often read-intensive, and may rely on latency-sensitive transfers, each consisting of small amounts of data. This marks a dramatic shift in how HPC storage systems are used. While some emerging read-intensive workloads may be able to rely on structuring within the data to construct efficient data retrieval plans based on caching or prefetching techniques, AI workloads and many data analytics routines are inherently required to access the data without any predictable ordering. According to the Department of Energy's 2020 AI for Science report (Stevens et al., 2020):

"AI training workloads, in contrast, must read large datasets (i.e., petabytes) repeatedly and perhaps noncontiguously for training. AI models will need to be stored and dispatched to inference engines, which may appear as small, frequent, random operations."

Figure 1 shows examples of prevalent access patterns for analytics, which are characterized by this lack of ordering.

Two technology trends have emerged as crucial to data-driven scientific discovery. First, the high-speed networks used within scientific computing platforms provide extremely low-latency access to remote systems, including billions of message injections per second and direct access to remote system memory via remote direct memory access (RDMA) operations. Second, solid-state disks (SSDs) accessed through the Non-Volatile Memory Express (NVMe) interface provide more than 1,000 times the performance of traditional hard disk drives for the small random reads used within data-intensive workloads. Interestingly, while HPC storage systems broadly leverage both high-speed networks and SSDs, this adoption was not driven by the need to provide low-latency access to remote storage, but by simulation requirements for fast point-to-point communication between processes and high-throughput requirements for access to HPC storage systems. With the advent of new read-heavy analysis workloads, however, low-latency remote storage access is now also a key enabling technology for new data-driven approaches to computational science.

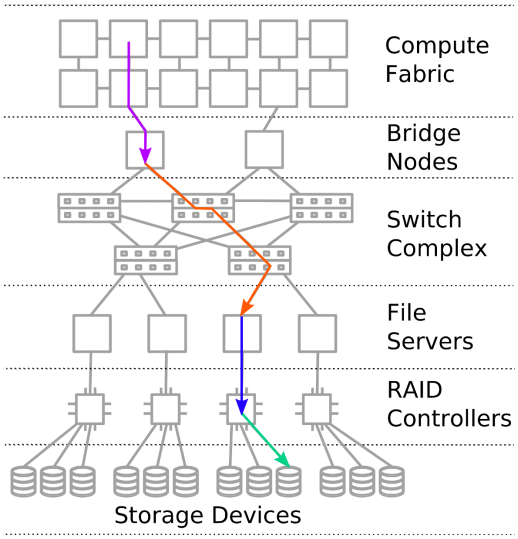

Figure 2: An exemplar disaggregated HPC storage architecture. Traditional HPC storage systems have been propelled by simulation workloads to optimize for aggregate bulk synchronous throughput. This is a key disconnect for data-driven analysis: systems designed to maximize aggregate throughput are poorly suited to individual random reads. Each access must traverse multiple distinct protocol hops, where each protocol hop has its own interrupt processing, buffering, handshaking, serialization, and access control conventions. These protocol translations were designed in an era when high-latency storage devices gated overall performance, an assumption that no longer holds today.

The evolution of HPC workloads has highlighted previously hidden shortcomings of modern storage systems. This is due to storage architectures that emerged in the early 2000s and remained static using storage servers with designated metadata and data roles, tightly attached to the HPC network, and focused on delivering throughput while relying on software layers that hid latency issues. Traditionally, system architects have relied on the increase of CPU frequencies and scaling out to prevent latency from affecting application performance. In recent years, however, CPUs have increased their computational power through the addition of CPU cores with decreasing frequencies that complicate real-time event processing. Scaling out to meet the IOPS requirements that modern HPC workloads place on the storage system is problematic as well. Balancing work to keep storage and network capability fully utilized is difficult at scale, resulting in underutilized resources and a higher total cost of ownership.

Emerging Challenges in HPC Storage

Modern HPC storage architectures were shaped by the performance characteristics of conventional hard drives. Conventional hard drives exhibit minimal (if any) onboard processing capability, low random access performance, and high latency. These characteristics placed a ceiling on overall storage system performance, and the remainder of the storage infrastructure was designed around mitigating their limitations as much as possible. Specifically, storage servers were designed to mediate all access to hard drives. By doing so, they could shape traffic (e.g., by serializing and batching), buffer data (e.g., through caching based on locality), and process I/O requests on more powerful host CPUs (e.g., by handling interrupts, packing and unpacking Remote Procedure Call (RPC) requests, and enforcing authorization) to make the most of hard drive capabilities. Hard drive access latency also had subtle implications for other elements of the storage system; there was no incentive to avoid latencies in the client-side operating system or the storage fabric as long as hard drives gated overall performance (**Figure 2**).

Low-latency access to storage

The architectural approach shown in Figure 2 was successful: it allowed HPC storage systems to extract maximum aggregate throughput from vast arrays of commodity hard drives. Limitations are evident, however, now that we attempt to match emerging IOPS and response-time-sensitive workloads to more capable low-latency storage devices. User-space APIs such as libaio or liburing can issue millions of operations per second from a single core, network interface cards can inject hundreds of millions of messages into a network per second, and these rates can be matched by just a few hundred NVMe storage devices. Despite these capabilities, modern storage servers are only able to process 100,000 RPCs per second from a single core. Even an incredibly high-end storage server with 100 high-frequency cores could service only 10 million read or write RPCs per second. Such performance strands over 90% of the network interface capability and saturates fewer than 10 fast NVMe devices.

In other words, the host-based RPC processing that in the past served to optimize access to storage devices has now become a hindrance. The server's ability to deserialize and process an RPC request and then serialize and send an RPC response is now the gating factor in the IOPS rate. The fastest RPC libraries, co-designed with high-performance interconnects and performing no server-side processing, have been unable to achieve even 500,000 RPCs per second per core. The traditional HPC solution of scaling out to achieve higher IOPS is inefficient; expanding the number of server CPU cores will increase complexity, footprint, and power demands, offer diminishing returns on aggregate IOPS rate, and effect no improvement in response time for individual accesses. The classic HPC storage architecture must now be revisited in the context of mixed workloads and the widespread availability of low-latency hardware components.

Scaling and maintaining low latency

Science teams driving these data intensive activities are pushing the scalability of their computations just as teams with simulation codes have before them, and it is paramount that storage systems support that scalability. Traditional caching and prefetching are not generally effective for these algorithms, eliminating a common option for accelerating access. On the other hand, the HPC networking community has learned much that can be applied to next-generation storage systems. Limiting the state associated with connections is an important enabler for scale-out, especially when there's no obvious structure in the communication as there is in many scientific codes.

Devices supporting protocols that require connection establishment are incredibly challenging to employ at HPC scale, but unfortunately, that is the current direction of network-accessible device protocols such as NVMe-oF. Connectionless models of communication have been demonstrated in HPC (Barrett et al., 2012) and supported in production hardware (Derradji et al., 2015): it is up to HPC to invent the fast, direct access to remote storage devices that will be a key enabling technology for scalable storage systems. HPC platforms have similarly been at the leading edge of requirements for high concurrency, low-latency access to remote memory, and extending proven techniques to enable similarly parallel and low-latency access to storage is a natural research direction. Alterations and alternatives to existing data transport methods for storage—perhaps built using compute-enabled devices—should be investigated and their potential demonstrated. User-land access to resources has also been shown as critical for maintaining low latencies, which will be critical in the data plane if not also in at least some aspects of the metadata plane. Approaches along these lines have begun to be explored in the larger storage community (Chen et al., 2021) but must be adapted to the scales and networks of HPC.

Securing access to storage devices

In addition to providing efficient access to storage devices, storage system software is also tasked with providing access control to the data stored within high-performance storage systems. In the current server-mediated access to storage model, the system software is tasked with enforcing all data access controls. As we move to a storage access paradigm that supports faster, low-latency access to storage devices, a

server-mediated access control scheme becomes a bottleneck that paralyzes emerging workloads rather than acting as a useful enforcement mechanism. At the same time, storage devices have gained richer interfaces and capabilities, including zoned namespaces (ZNS) and embedded functions in the form of computational storage, and thus it is clear that security models that treat storage devices as only a repository for stored data are obsolete.

More direct access to storage devices from large numbers of client processes, which may include user-space access to remote storage devices, must provide new models of security not currently provided by either the network protocols or storage devices. While the NVMe standards body has defined multiple methods for securely accessing storage, none of these mechanisms are currently a good match for data-intensive scientific discovery. The two most common NVMe security methods, in-band authentication and per-request security, are focused on ensuring that clients are authenticated with servers but cannot differentiate between data plane operations that read data or write data and control plane operations that create or destroy on-device namespaces. And while key-per-IO is a novel model that enables every disk access to be secured separately, the overheads of checking an encryption key for every operation is antithetical to low-latency access to storage devices. Instead, new security models that expose the performance advantages of zoned namespaces (Bjørling et al., 2021) and leverage scalable approaches to embedded compute, such as computational storage and SmartNICs (Li et al., 2020), require additional research.

Enabling a Future for Data-driven Science

A great deal of effort was required to stabilize HPC storage and make it trustworthy, but it did happen. Multiple production file system options exist for data centers to choose from, and checkpoint and restart for HPC codes has largely been addressed. But storage system designers cannot rest on their laurels, and storage is not a solved problem. Even more than for simulation codes, the potential benefits of HPC for AI and analysis applications hinge on high performance storage. We need not just innovation, but innovation that goes hand in hand with these scientific objectives.

Architecturally, the community must revisit the data path between analysis applications and storage devices. In much the same way that user-space RDMA access has revolutionized HPC networking (removing handshaking, buffering, and host processing from the interprocess communication path) and allowed networks to keep pace with memory throughput, we must adopt new HPC storage access paradigms that minimize obstructions in the storage data path and allow storage systems to keep pace with NVMe capabilities. The need for RPC processing can be minimized (by thoughtful partitioning of work to control planes), any remaining RPC processing or asymmetric transfer can be offloaded to smart devices, and the complete data path can be holistically evaluated to eliminate duplicate and superfluous protocol translations that collectively leach latency from the system.

From a device interface perspective, storage systems traditionally divide responsibility between the storage device and host rigidly: the device is responsible for handling data block updates, and the host is responsible for data processing. But as SSDs continue to replace hard disks at the front-line storage tier, block interface support requires complex firmware that affects device performance and cost, and as storage becomes disaggregated from computation, reducing data movement between the device and host becomes crucial. Novel interfaces, like Zoned Storage, have emerged to reduce firmware complexity by delegating responsibilities to the host, and computational storage allows data to be processed on the device in accordance to application needs, blurring the divide between device and host. Future work will need to adapt popular application types to fully leverage the capabilities of these devices and explore the right balance of near-storage computation for different tasks.

User abstractions are another key piece of the puzzle. Building fast and productive storage systems will require not only addressing these technology challenges but also understanding emerging science needs. The HPC storage community has contributed interface advances in the past, including concepts eventually

adopted in the mainstream, but recently most storage abstraction innovation has occurred elsewhere, with cloud service providers offering options such as column stores, document stores, key-value stores, streaming data infrastructure, and object stores. HPC storage researchers must work together with technology providers and domain scientists to find abstractions that match science needs and then to develop scalable storage services embodying those abstractions.

The HPC storage research community also a needs to be reinvigorated. A misconception persists that HPC storage is a solved problem: new storage systems are iteratively designed and deployed by solving formulas based on commodity market forces and logistical constraints. In reality, however, many unsolved problems remain in high-performance storage, especially as high-performance storage comes to the forefront as the key to enabling both simulation and data-driven analytics use cases. The high-performance storage community must innovate within this space and then translate those innovations into solutions for our data-driven science partners. HPC storage and its workloads must become first-class citizens within computer science curricula, coordinated research thrusts, and partnerships between industry, academia, and governments.

Conclusions

The push to achieve the largest and most complex scientific discoveries using high-performance computing requires heroic efforts from computational scientists, computing system designers, and software developers. But critically, these tremendous efforts have proven to successfully flow downstream and make equally important, but less computationally demanding, scientific discoveries tractable. By design, a calculation that was entirely heroic a decade ago, can now be achieved by a handful of highly motivated graduate students. To usher in this same downstream effect for data-driven science, a set of sustained and heroic efforts are needed for building and operating storage systems that can support highly concurrent and low-latency access to massive volumes of scientific data. With this key underpinning under development and then in use, we enable the additional efforts needed to extract new insight and invent new methods for accelerating data-driven scientific discovery. And in several years, as the benefits of new methods for analyzing data are realized and made commonplace, small teams of highly motivated graduate students will perform data-driven searches for discovery that could not be dreamed of as possible within contemporary HPC data centers. The road ahead, and its inevitable roadblocks and detours, will be difficult and surprising, but the rewards at the end of this journey are too great to resist.

Acknowledgment

This work was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

Author Bios

Bradley Settlemyer is a senior scientist in Los Alamos National Laboratory's HPC Design group. He received his Ph.D. degree in computer engineering from Clemson University in 2009 with a research focus on the design of parallel file systems. He currently leads the storage systems research efforts within Los Alamos' Ultrascale Research Center and his team is responsible for designing and deploying state-of-the-art storage systems for enabling scientific discovery. He is the Primary Investigator on projects ranging from ephemeral

file system design to archival storage systems using molecular information technology and he has published papers on emerging storage systems, long distance data movement, system modeling, and storage system algorithms. Contact him at bws@lanl.gov.

George Amvrosiadis is an assistant research professor of electrical and computer engineering and, by courtesy, computer science at Carnegie Mellon University, Pittsburgh, and a member of the Parallel Data Laboratory. His current research focuses on distributed and cloud storage, new storage technologies, high-performance computing, and storage for machine learning. He co-teaches courses on cloud computing and storage systems. Contact him at gamvrosi@cmu.edu.

Philip Carns is a principal software development specialist in the Mathematics and Computer Science Division of Argonne National Laboratory, Lemont. He is also an adjunct associate professor of electrical and computer engineering at Clemson University and a fellow of the Northwestern-Argonne Institute for Science and Engineering. His research interests include characterization, modeling, and development of storage systems for data-intensive scientific computing. Contact him at carns@mcs.anl.gov.

Robert Ross is a senior computer scientist at Argonne National Laboratory, Lemont, and a senior fellow at the Northwestern–Argonne Institute for Science and Engineering at Northwestern University, Evanston. Dr. Ross's research interests are in system software and architectures for high-performance computing and data analysis systems, in particular storage systems and software for I/O and message passing. Rob received his Ph.D. degree in computer engineering from Clemson University in 2000. Rob was a recipient of the 2004 Presidential Early Career Award for Scientists and Engineers. Contact him at rross@mcs.anl.gov.

References

AI for Science. (2020). Office of Scientific and Technical Information (OSTI). https://doi.org/10.2172/1604756

The Portals 4.0 network programming interface. (2012). *Sandia National Laboratories*.

The BXI interconnect architecture. (2015). *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, 18–25.

Scalable Persistent Memory File System with Kernel-Userspace Collaboration. (2021). *19th {USENIX} Conference on File and Storage Technologies ({FAST} 21)*, 81–95.

ZNS: Avoiding the Block Interface Tax for Flash-based SSDs. (2021). *Proceedings of the 2021 USENIX Annual Technical Conference (USENIX ATC'21)*.

Leapio: Efficient and portable virtual nvme storage on arm socs. (2020). *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 591–605.