PURESTORAGE®

TECHNICAL WHITE PAPER

Pure Storage FlashBlade and Ethernet for HPC Workloads

Countering the myths of HPC and parallel file systems

Contents

Introduction

This white paper discusses how NFS with Pure Storage® FlashBlade® and Ethernet delivers high performance and data consistency for high performance computing (HPC) workloads. The paper also shows how FlashBlade out-performs parallel file systems, which are commonly used for HPC. The companion document "Toward a More Simple, Scalable HPC Storage Model" discusses how FlashBlade is enterprise-ready and handles storage requirements for the modern HPC-AI workloads.

The HPC Technical Ecosystem

HPC clusters involve a group of powerful computers that are grouped together to perform parallel tasks to solve complex computational problems at very high speeds. An HPC cluster can consist of bare metal servers, virtual machines (VMs), or as microservices in a Kubernetes cluster as shown in Figure 1. AI-augmented HPC (HPC-AI) provides more accurate results during the simulation and measurement cycles while processing and analyzing large datasets. A high volume of data is generated during the modeling and simulation of complex scientific models for various computer aided engineering (CAE) and industry-specific custom applications.

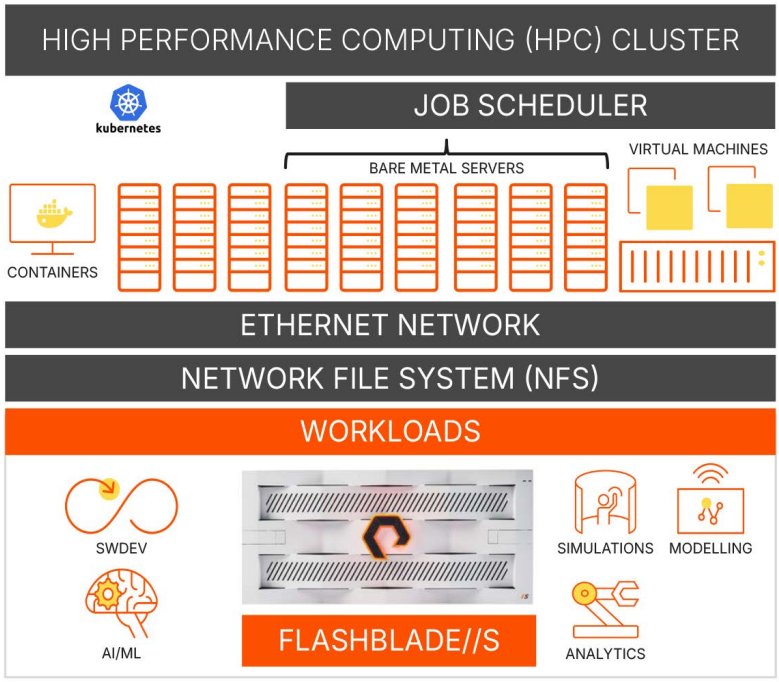

FIGURE 1 A high performance computing (HPC) Cluster with high-speed flash storage.

Pure Storage FlashBlade//S: High-performance Storage Parallelism for HPC

Pure Storage FlashBlade//S™, with its DirectFlash® modular architecture provides a high-performance storage solution for traditional HPC and modern AI-augmented workloads to scale performance and capacity on-demand, with superior power efficiency. It is purpose-built, and delivers unstructured data via NFSv3 and NFSv4.1, and is superior over conventional NFS servers built on a Linux workstation.

The performance advantages of FlashBlade//S systems begin with a highly-parallelized distributed database for metadata that can handle a high throughput of millions of small files and high numbers of large files concurrently in the same platform without any additional configuration and tuning. Additionally, managing and integrating the FlashBlade//S into an HPC workflow using APIs is simple and easy.

NFS on FlashBlade is architected to handle concurrent read and write requests between the clients and the server at a higher speed than traditional file services hosts. Unlike a parallel file system, NFS on FlashBlade allows the scaling of CPU and GPU independently with flash storage. The integrated data management and data protection capabilities of FlashBlade also make the NFS enterprise ready.

FlashBlade NFS over Ethernet is widely used in many commercial and research HPC vertical workloads. It supports both traditional transport control protocol (TCP) and remote memory direct access (RDMA) for high-speed data transfer between it and the NFS client. However, there has been a constant debate whether to use NFS over Ethernet instead of parallel file systems with Infiniband.

Tackling the Myths of NFS Deficiencies

NFS over Ethernet on FlashBlade is a superior solution to leveraging a PFS using "just a bunch of disks" (JBODs) with InfiniBand (IB) for HPC workloads.

Myth #1: A PFS Is the Most Suitable Infrastructure for HPC Workloads

PFSs have been successfully leveraged in many HPC environments in the past due to their ability to scale and to process IO in parallel on large file sizes. As shown in Figure 2, the metadata server (MDS) keeps track of the metadata operations while the PFS breaks the files into objects that are written to the object storage targets (OST) in blocks. The PFS strips a single large file into smaller chunks and uses the XFS file system to write it to the underlying disks. Many times metadata performance with a PFS is a bottleneck and the complexity of implementing parallelism introduces more points of failure, leading to data integrity challenges. Manageability overhead and the complex troubleshooting process with a PFS can become disruptive for HPC workloads in production.

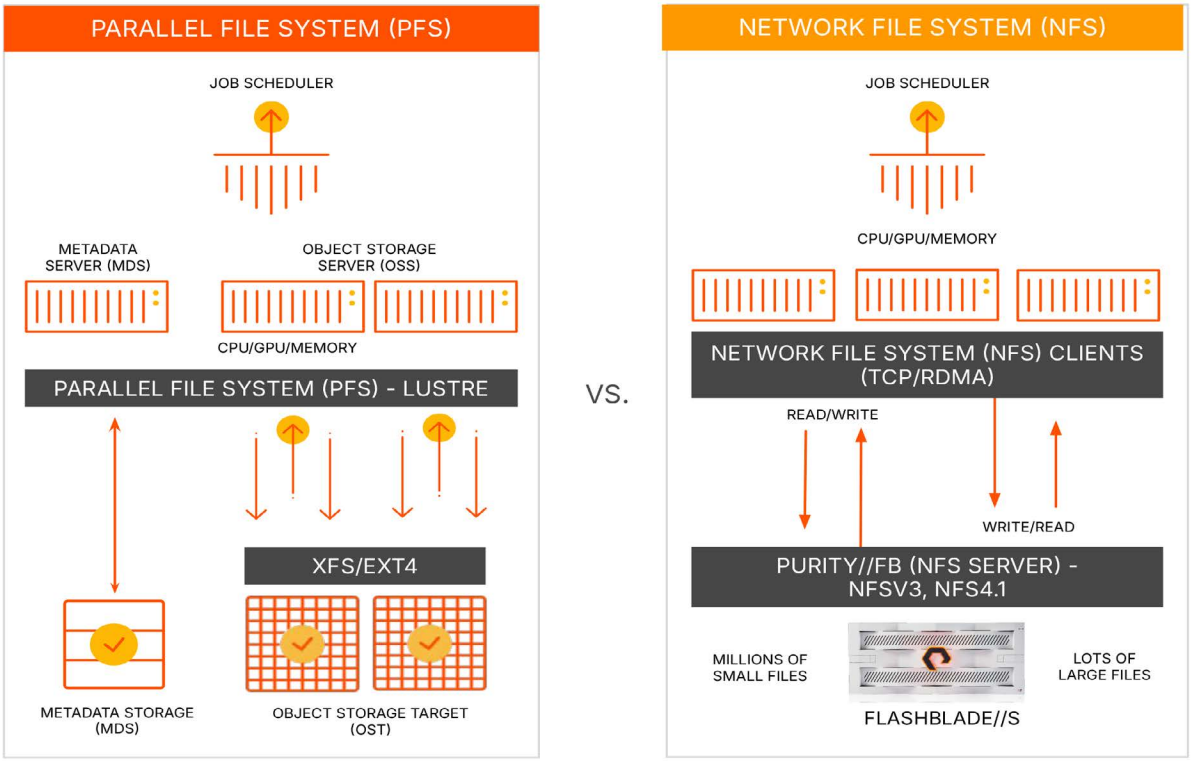

FIGURE 2 A high-level layout of a parallel file system such as Lustre in comparison to a network file system on a FlashBlade system.

NFS is a file transfer protocol and the Purity//FB storage software in FlashBlade//S is responsible for all the parallel read and write operations, while metadata is distributed across all blades in the storage array. NFS has gone through rigorous standardization over the years and has wide adaptability in HPC workload segments. NFS on FlashBlade supports small and large IO sizes where writes are coalesced in segments before being written to backend flash. NFS on FlashBlade is designed as a simpler architecture, has fewer moving parts in the fault domains, and very high uptime to provide high availability data access with very little management overhead.

Myth #2: Unlike a Parallel File System, NFS with FlashBlade Is Not POSIX Compliant

POSIX standards were originally designed for local file systems like XFS and EXT4. However, these file systems were not shareable over the network and were also not scalable. Parallel file systems often offer strong data consistency and read/write atomicity to the backend storage. And while a PFS provides a distributed locking mechanism that prevents the clients from writing to the same portion of the file simultaneously, recovery from data loss can be challenging and time consuming due to lack of fault tolerance and redundancy at the storage level.

NFS was never designed to be fully POSIX compliant, but it has morphed to conform to the standard as much as possible. However, NFSv3 is still widely used for performance reasons. A modern Linux kernel provides robust close-to-open processes (CTO) to maintain file consistency, synchronous metadata and data transfers to FlashBlade, reducing the risk of data loss and using locks to manage access on file(s) for single writers with multiple readers or concurrent read and write operations. FlashBlade handles the atomic commit and consistency once it receives the data from NFS and sends an acknowledgement to the client. Data resiliency with erasure coding makes NFS with FlashBlade mean time to data loss very high.

Myth #3: Parallel File Systems Have Better Performance Than NFS on FlashBlade

A PFS has always been associated with highly scalable performance, with low latency using RDMA and Infiniband for read/write IOs to backend block storage. A PFS has the ability to create, list and delete directories and files at a very high speed. But they were originally designed for traditional hard disk geometries that perform well with larger IO sizes from 1MB to 16MB. HPC workloads with small files requiring higher IOPs is still a challenge with Parallel File Systems. New, cost efficient solid State devices (SSDs) with quad level-cell (QLC) technology have replaced spinning media in recent times. The native PFS characteristics have not changed much, thus leading to write amplification issues and high wear level as SSDs start to age, which degrades the performance over time.

NFS differs from a PFS mainly in the data path, how data is read and written to the backend storage from a performance perspective. Even though FlashBlade supports a 512k IO size by default, the NFS client can negotiate the IO size over TCP or RDMA with FlashBlade (NFS server) where data packets are coalesced to fewer and larger writes to flash storage for various file sizes to minimize the flash wear level. NFS on FlashBlade can parallelize and evenly distribute the load over multiple TCP & RDMA connections for HPC applications, leading to high-bandwidth operations under low latency. Metadata operations for small files on FlashBlade do not require any additional configuration or tuning for improved performance. NFS can create, list, and remove directories and files three to five times faster than PFS at scale. FlashBlade using QLC flash has demonstrated disaggregative scale with respect to high performance and capacity compared to PFS.

FlashBlade NFS provides additional value for HPC environments, apart from the list of benefits stated above.

1. **Security:** FlashBlade supports mode bits (read/write/execute) and NFSv4.1 access control lists (ACLs) with granular security over POSIX ACLs for directories and files for user authorization. NFS and Kerberos provide stricter authorization mechanisms for users to access filesystems on FlashBlade. Data at rest and in motion (replication) is automatically encrypted. User authentication using LDAP also adds another layer of security for HPC users accessing data from NFS.

2. **Sustainability:** FlashBlade delivers 15% lower energy consumption than a leading Parallel File System storage vendor using QLC flash providing similar performance per rackspace. Lower energy consumption by Pure Storage allows lower cooling requirements to scale more CPU/GPUs and diligently manage the power requirements in the datacenter.
3. **Error handling:** An NFS client has the ability to retransmit a data packet to the server after a predefined time out period if the data packets are lost during transport over a lossy network. The data resiliency capabilities in FlashBlade allow rapid recovery of data in case of hardware failures on the storage. In comparison, a PFS has an error checking mechanism but the distributed nature of the data adds complexity, making error detection more challenging.

Ethernet vs. Infiniband for a High-performance Computing Network

Infiniband (IB) connectivity between compute nodes and storage along with switches from Mellanox is widely used as a high-speed network for HPC workloads over Ethernet. However, Ethernet has evolved very rapidly in recent times and is continuing to grow as traditional HPC workloads are transitioning to HPC-AI. NFS on FlashBlade uses Ethernet as a standard network between compute nodes and storage expansions.

This section will compare Ethernet with Infiniband and understand the recent changes to Ethernet network architectures that are comparable and sometimes better than Infiniband. Pure Storage has a two-fold recommendation:

1. New HPC-AI implementations can use high-speed Ethernet connectivity with low latency for inter-node communication and data transfer to FlashBlade.
2. Existing HPC users can continue using IB for the inter-node communication in the cluster and use Ethernet between the compute cluster nodes and FlashBlade for high-speed file transfer to storage.

The following are some high-level concerns that come up during an Infiniband vs. Ethernet conversation.

1. **Performance:** IB networks and switches provide high throughput and low latency to HPC applications and workflows over MPI-IO and RDMA. IB is capable of providing network speed up to 400 Gbps (NDR) that provides the massive data transfer required for HPC-AI workloads.

   However, Ethernet has also caught up to high-speed data transfer and can provide speed up to 800 Gbps under low latencies. Ethernet switches no longer block the data packets from inefficient queues leading to congestion. The packet spraying feature in modern Ethernet switches distributes data packets on multiple paths independently to avoid congestion. The Ultra Ethernet Consortium (UEC) is establishing new standards to handle data transfer with speed upwards of Tbps. Pure Storage is a member of the UEC.
2. **Reliability.** IB networks have proven to be reliable, with better congestion control and flow control mechanisms to avoid any packet loss during high-throughput data transfer under low latency. Infiniband has lower protocol overhead with RDMA by offloading network processing from CPUs to free up computational resources for high-performance tasks.

   In the past, NFS over TCP and Ethernet switches was constrained by how TCP handled the loss detection and out-of-sequence packet arrival from the sender and the receiver. Modern TCP implementations are more resilient to out-of-order packets and the newer Ethernet switches can reorder packets in the flow for an in-order delivery. NFS on FlashBlade supports RDMA along with TCP. Tighter resiliency at the TCP layer and reduced tail latency with modern Ethernet switches like NVIDIA Spectrum 4 SN5600, Broadcom Tomahawk 5, and Cisco Nexus 9000 series (to name a few), makes NFS with FlashBlade the best alternate choice to IB for HPC-AI workloads.

3. **Scalability and error handling:** A PFS with IB introduces more points of failure due to the concept of parallelism. IB provides low latency and higher bandwidth but error recovery times are long and complicated due to the high-speed nature of the connection. Scalability with the IB network is limited to spine and leaf layout as the IB network is designed to provide uniform performance with efficient load balancing.

   Modern Ethernet switches on the other hand use error-checking mechanisms like cyclic redundancy check (CRC) to detect data corruption in packets to ensure consistent error handling capabilities across the network implementation. Modern high-speed Ethernet switches support higher radix (port density) to provide a simple network design with fewer network hops when the network scales beyond a two-tier spine and leaf layout. NFS with FlashBlade and Ethernet has standardized robust error detection and retry mechanisms to ensure data consistency and integrity.

Conclusion

The following are the key takeaways from using NFS with FlashBlade over Ethernet in comparison with a PFS and IB for HPC-AI environments:

- It's a more future-proof solution and provides a better return on investment (ROI) for high-speed networking and storage.
- It's simple to deploy and manage the network and storage infrastructure and doesn't require a highly specialized skill set.
- It delivers high performance under low latency, and the reliability needed for high-speed data transfers.
- It uses less expensive hardware that is more ubiquitous than Infiniband without any vendor lock-in.

Additional Resources

- Learn more about FlashBlade//S.
- Speed up file handling tasks with the RapidFile Toolkit.
- View all the Pure Storage AI and machine learning blogs.

About the Author

Bikash Roy Choudhury is a director at Pure Storage with 30 years of industry experience. His primary focus areas are in high growth HPC industry verticals including electronics design automation (EDA), genomics, research, manufacturing, and finance in on-premises, hybrid/public cloud environments. He has also worked on various solutions with Kubernetes, software development and monitoring tools using RESTful APIs, and integrating them with data platforms in hybrid and public clouds. Bikash has also worked closely with strategic partners like Azure, Perforce, GitHub, JFrog, Illumina and Nvidia. In his current role, he is driving strategies and business initiatives to win in the high growth HPC market segment.

PS2700-01-en 11/24