PURESTORAGE®

TECHNICAL BRIEF

FlashBlade//EXA

Storage optimized for artificial intelligence at scale

Contents

Abstract

After decades in a research niche, Artificial Intelligence (AI) has become a mainstream information technology (IT). Enterprises in finance, medicine, the military, logistics, and many other fields use it to deliver business value.

This has created a need for AI-capable IT systems—"AI-capable" because AI's computing, network, and storage needs are well beyond those of conventional enterprise applications.

For example, to train an AI model to produce useful inferences, thousands of specialized computers (GPUs) process petabytes of data delivered to them at rates of terabytes per second (TB/s). And as models become larger, their needs will only increase.

To support *high-performance computing* (HPC), a branch of IT with storage and I/O needs similar (but not identical) to those of AI, the industry has developed *scale-out parallel storage systems*. These have sometimes been adapted to satisfy AI requirements.

The Pure Storage® **FlashBlade//EXA**™ architecture is specifically designed for AI (and HPC) deployments. It consists of a *metadata cluster* based on FlashBlade® technology that integrates with user-provided *data nodes* and networking to deliver multiple TB/s of throughput with commensurate metadata service to AI and HPC clients. This brief describes how **FlashBlade//EXA** delivers efficient, easy-to-deploy, scale-out storage with the capacity, throughput, and metadata performance that modern AI and HPC demand.

The artificial intelligence tsunami

AI, particularly in its *generative* form (GenAI), has become a mainstream technology with adopters using it in increasing numbers of applications. IT service providers, both *hyperscalers* like Microsoft and Google, and newer entrants specializing in AI, are enabling enterprises in many fields to use AI to offer new services, increase efficiency, and reduce operating costs.

Today, many enterprises meet their AI needs via service providers that cater to the growing market for model development and production use. But as has happened with other enterprise applications, technology-aware users are moving their AI applications in-house to improve predictability, performance, flexibility, information security, and cost effectiveness.

AI workflows

Figure 1 represents a typical AI processing workflow.

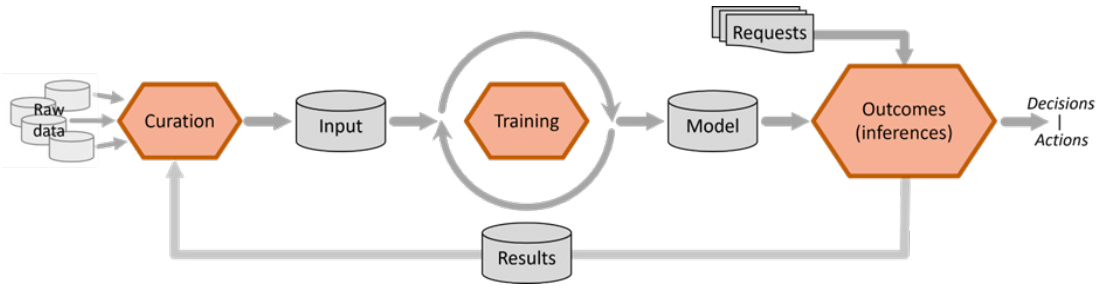

FIGURE 1 Representative AI workflow

An AI project starts with raw data, typically from multiple unrelated sources. Curation eliminates irrelevant items and transforms the rest for use in iterative *model training*. Data scientists test multiple candidate models, eventually converging on one that produces useful *inferences* (results) reliably. Trained models transition to production, where they respond to requests that support decision making. As the figure suggests, AI workflows are cyclic. Models are retrained frequently, using production feedback and adding parameters and new data sources.

Data curation, training, and inferencing are highlighted in Figure 1 because they all need fast access to huge amounts of data. For example, during training runs that may last a week or more, thousands of GPUs process petabytes of input data and produce terabyte-size *checkpoints* as often as every few minutes. AI deployments need storage systems that can (a) deliver data at rates that keep GPUs productive, and (b) absorb frequent large checkpoints. And as "tier 1" applications that influence enterprise operations and decision making, production models must be highly available and responsive to ad hoc requests and process-driven inputs.

AI vs. enterprise IT

AI's computing, network, and storage needs differ from those of typical enterprise applications in three respects:

Processing

Enterprise storage systems typically serve one or a small number of applications or provide virtual desktop infrastructures (VDIs). AI models are trained by *computing clusters* of hundreds of servers, each hosting multiple GPUs and constantly accessing storage.

Data transfer

Most conventional application I/O needs can be met by today's enterprise storage systems. AI training, however, can require over two gigabytes per second of input data for every GPU—well beyond the capability of even the most powerful enterprise storage systems.

Storage services

For enterprise applications, data *durability* (protection against loss) and *availability* (accessibility by clients) are usually the top storage priorities. For AI deployments, I/O performance, capacity, and availability are the top priorities. Some enterprise features—snapshots and replication, for example—are not feasible in large AI deployments.

A model for AI storage

Eliminating bottlenecks is key to meeting AI's massive I/O throughput needs. But it's not just about throughput. The metadata access required to create, retrieve, and manage the billions of data items used in typical AI processes can constrain overall system I/O performance.

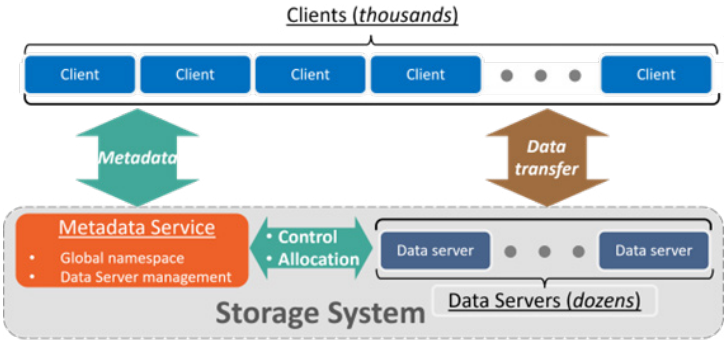

FIGURE 2 Parallel (disaggregated) storage system model

To minimize bottlenecks, AI storage systems typically implement some form of the *parallel storage system* model shown in Figure 2. The essence of the model is *disaggregation* (separation) of metadata and data and use of separate network paths for accessing them. In this model, metadata service and data servers have distinct roles:

Metadata service

Manages capacity and I/O load balance for the overall system and presents a global (system-wide) data item namespace to clients. The service maps (relates) item names in the namespace to the data server locations of their content. Clients access metadata to obtain the locations of existing data items and to allocate space for creating new ones.

Data servers

Provide direct client access to data item content. Each data server presents a separate local abstraction (e.g., file system or object bucket) to clients.

Given names in the namespace, metadata services return data servers' local identifiers (e.g., file handles or objectIDs) for item contents. Clients use the local identifiers to address item contents in read and write operations. The metadata service is not in the data transfer path.

Benefits of disaggregation

Parallel storage systems are inherently *scale-out*—they expand by the addition of resources (data servers and network bandwidth) unlike enterprise systems that typically *scale up* by the substitution of more powerful components, so their potential for growth is significantly greater.

In principle, parallel systems allow data and metadata capacity and performance to scale independently. HPC deployments that mainly read and write large items sequentially can configure data servers and network bandwidth as needed without having to add unnecessary metadata capacity. AI deployments whose workloads include billions of small item accesses can increase metadata resources without having to configure data capacity they can't use.

Parallel storage systems aren't easy

The first parallel storage system implementations were either proprietary (e.g., GPFS[1]) or open source (e.g., Lustre[2]). The systems were designed to support the large file access patterns commonly found in simulation, exploration, experimentation, and other HPC applications. They were optimized for high aggregate data throughput scenarios in which modest numbers (dozens) of clients read and write entire files concurrently. They achieved high performance largely by disaggregating metadata and data and scaling the latter. Throughput could increase as needed by the addition of storage capacity and network bandwidth, while OPEN, GETATTR, and similar metadata-only operations would address separate devices on separate network paths and were therefore not delayed or blocked by large in-progress data transfers.

But disaggregating metadata and data doesn't automatically enhance metadata performance. Metadata services must also be scalable, especially in AI deployments, where rapidly changing I/O workloads intermix large item transfers with billions of metadata and small item accesses. Systems designed for HPC workloads may not adapt well to AI deployments that need to retrieve or store hundreds of thousands of small items per second on behalf of thousands of GPU clients while simultaneously reading and writing large items.

Supporting parallel storage systems

Enterprise storage systems consist of controllers, storage devices, network interfaces, and software, all designed to work together. Users typically specify performance, capacity, and resilience needs and vendors propose systems that meet them "out of the box."

Early adopters of parallel storage systems had to acquire, configure, manage, and upgrade the components that made up their systems. The highly customized environments that resulted required significant technical expertise and support resources, typically provided by dedicated in-house teams. Although expensive to manage and evolve, over time, custom storage systems supported by in-house teams came to be regarded as the norm for HPC deployments.

Early implementations

Early parallel storage systems could not utilize mature file system interfaces. The common data access methods—POSIX, NFS, and SMB—were (and still are) designed for file systems whose data and metadata are co-located in the same server. They make no provision for disaggregating data and metadata or for clients to access data through multiple data server mount points. Parallel storage system interfaces had to be developed "from scratch.".

Developers of these early systems created data access protocols as well as custom client software. Their customers typically maintained dedicated teams to install, configure, and maintain hardware and custom software, both on computing clients and on storage system components. Over time, investments in these systems and the expertise to operate and maintain them encouraged *vendor lock-in*—change was unaffordable.

Variations on the parallel storage system theme

Some vendors developed minor variations of the parallel storage system model, one of which is illustrated in Figure 3. The figure represents an architecture with a number of *stateless* metadata servers. Data servers hold both data (on low-cost media) and metadata (on higher-performing media). In principle, the architecture allows independent data and metadata scaling, but persisting and retrieving metadata increases network traffic. Moreover, configuring, managing, and scaling such systems is complex, leading some users to avoid change by greatly over-provisioning at initial system installation, resulting in inflexible configurations and high system cost.

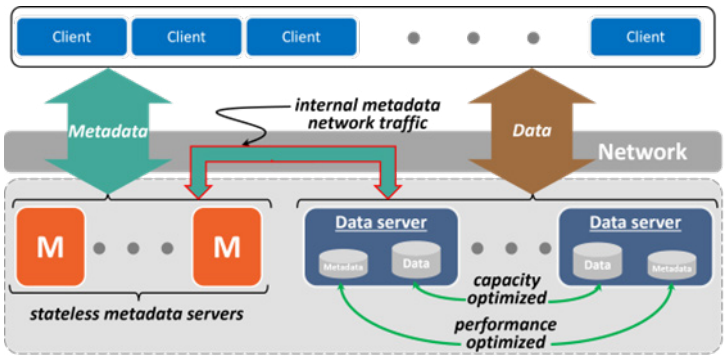

FIGURE 3 Alternate disaggregated architecture

The emergence of standards

Driven by the limitations of proprietary and open source-based parallel storage systems, the *Internet Engineering Task Force* (IETF) ratified the NFSv4.1 standard in the early 2000s. NFSv4.1 included a *parallel Network File System* (pNFS) protocol. The standard, since evolved to NVFv4.2, includes features[3] to allow for data servers that present either file, object, or block abstractions to clients, but it does not define the client-data server protocols per se.

In 2015, the FlexFiles[4] standard for data servers that host local file systems was introduced. Today most pNFS systems utilize FlexFiles, although object-based systems are starting to emerge. Clients of NFSv4 parallel file systems use the pNFS protocol for metadata interactions and the NFSv3 protocol to read and write item contents on data servers' local file systems.

Newly developed parallel storage systems are increasingly adopting the NFSv4 standard. Standards-based systems have advantages for both users and storage system vendors:

- All enterprise server operating systems support NFSv4 protocols, including pNFS. Client-side support for storage is therefore an operating system feature rather than being provided by storage vendor custom software. Similarly, all potential data server operating systems support server-side NFSv3. Standards help keep users' overall support costs manageable and avoid vendor lock-in.
- Standard interfaces broaden product content options for vendors who supply both metadata and data servers, and encourage potential customers, many of whom have policies of avoiding vendor lock-in. For vendors who offer pNFS metadata servers to be combined with third-party data servers it increases addressable market.

Incumbent vendors have significant installed bases of their proprietary systems, so most continue to adapt open-source solutions by adding custom protocol and management software to simplify operations somewhat. But custom solutions remain a support cost and availability issue as users refresh client hardware and update storage system hardware and software.

Pure Storage's parallel storage system, FlashBlade//EXA, is based on the NFSv4.2 and FlexFiles standard features that enable parallel storage systems.

FlashBlade//EXA: Data storage for AI

A **FlashBlade//EXA** system consists of a Pure Storage-supplied scale-out *metadata cluster* based on FlashBlade technology integrated with user-supplied commodity data servers (called *data nodes* in Pure Storage documentation). Figure 4 illustrates the system's structure.

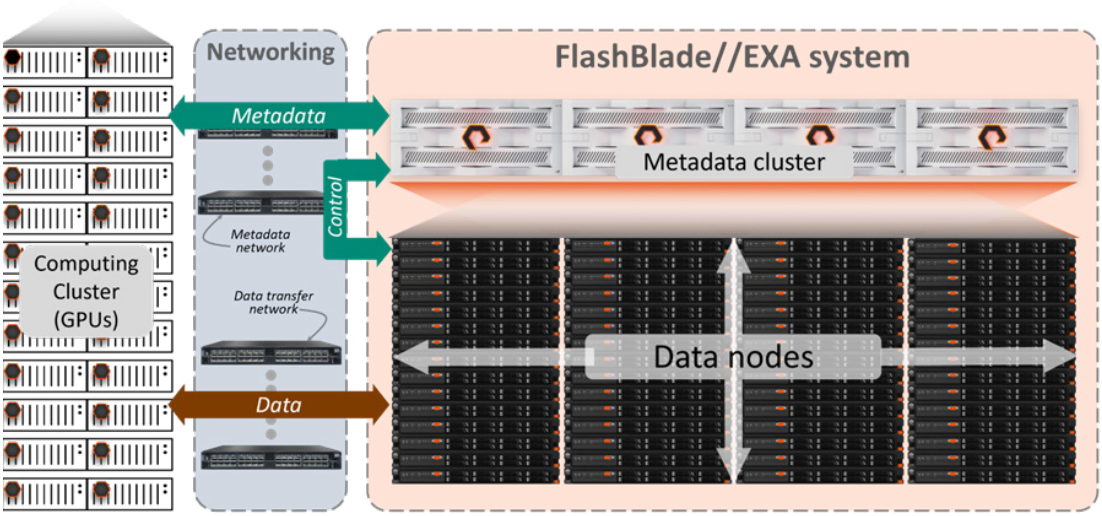

FIGURE 4 FlashBlade//EXA system architecture and communication paths

Hardware

The easily expandable metadata cluster hardware consists of one or more FlashBlade chassis based on user needs. The cluster's Purity software includes server-side pNFS protocol support developed by Pure Storage and *control plane* software. **FlashBlade//EXA** metadata clusters inherit the proven robustness and performance of the underlying FlashBlade architecture.

FlashBlade//EXA users provide the data nodes for their systems. These can be any type of commodity server that meets the basic requirements summarized in Table 1. Pure Storage expects some users to repurpose existing server "farms" as data nodes.

Component	Requirement
Processor	X86, 32+cores, 192GB DRAM
Storage (SSDs)	NVMe (PCIe Gen4+), 3.8-61TB
Network	400Gb Ethernet interfaces (x2)

TABLE 1 Data node requirements

Networking

Metadata and data traffic are carried on (conceptually) separate networks. All metadata clusters include dual *eXternal Fabric Modules* (XFMs) that are used for (a) inter-chassis communication within the cluster and (b) connecting to users' backbone networks for client access to metadata. Pure Storage does not supply the network facilities for client–data node communication.

Metadata cluster software

FlashBlade//EXA systems use standards-based NVFv4 parallel file system protocols rather than proprietary or open-source solutions.

Metadata clusters run the same Purity software as other FlashBlade family members, augmented by pNFS for metadata interactions. Figure 5 shows the principal software components:

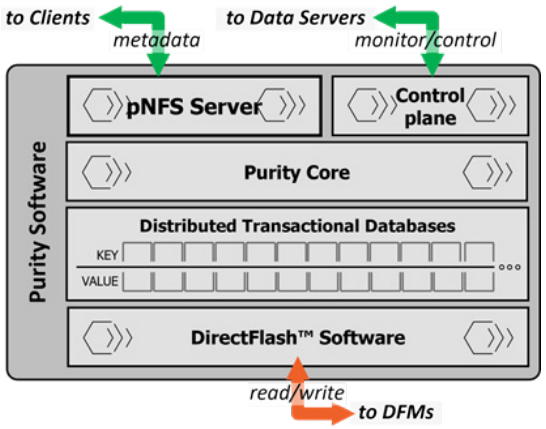

FIGURE 5 Purity software

PNFS protocol

Implements the pNFS server-side protocol for interactions with clients and data nodes.

Purity Core

Schedules, load-balances, and manages interactions with clients and data nodes.

Distributed transactional databases

Organize system flash and NVRAM as a set of distributed key-value databases. The database organization is an major contributor to the system's high metadata performance.

DirectFlash software

Manages the DirectFlash™ Modules (DFMs) that provide persistent storage for metadata and cooperates with DFM firmware to read and write flash and NVRAM with almost none of the internal data movement and write amplification typical of off-the-shelf SSDs.

Control plane

Monitors data node status and utilization, allocates storage for items created by clients, balances data node utilization, and generates alerts as needed.

FlashBlade//EXA systems use Purity's key-value databases to organize the metadata used to locate data item content on data nodes. Keys are names in the system's global namespace; values are item properties—location, size, access permissions, and so forth.

Data node software

FlashBlade//EXA software also includes **Purity//DN**, a complete data node operating suite with an executable system image, a zero-touch network-bootable installer, monitoring and visualization tools, and *playbooks* for common configuration and daily management tasks.

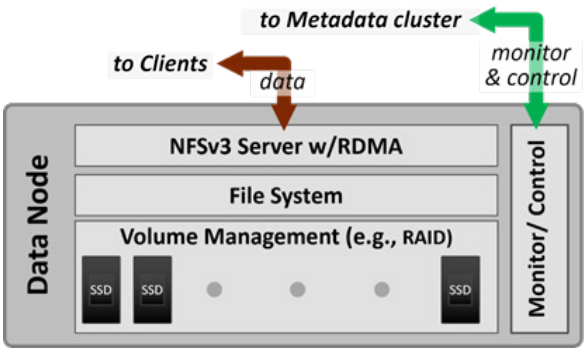

FIGURE 6 Data node software

Data node SSDs are user-configurable for mirroring (for optimal I/O performance) or single-parity RAID (for maximum usable capacity). Each data node hosts a local file system, whose data items clients read and write using the NFSv3 protocol with *Remote Direct Memory Access* (RDMA) support. The metadata cluster control plane manages overall data node configuration, allocates space for new items, and monitors data node status and utilization.

Workflow

FlashBlade//EXA systems utilize the pNFS FlexFiles standard protocol for data transfer. Each name in the global namespace corresponds to a file whose contents are located on one or more data nodes. Data nodes use the NFSv3 protocol to transfer data to and from clients.

To access existing data items, clients send NFSv4.2 file OPEN and similar commands to the metadata cluster The cluster verifies permissions, and for commands that result in data access, returns *layout* structures that contain item locations. Clients retrieve or modify data items by using NFSv3 to interact directly with data nodes.

To create new items, clients send NFSv4.2 CREATE commands to the metadata cluster. The cluster selects one or more data nodes based on utilization, creates an empty file on it, and sends the client a layout structure that indicates the file's location. The client uses NFSv3 to write item content directly to data nodes.

Metadata clusters and data nodes exchange status, performance, space inventory, and alert information, but most of a production **FlashBlade//EXA** system's network traffic consists of direct data transfers between clients and data nodes.

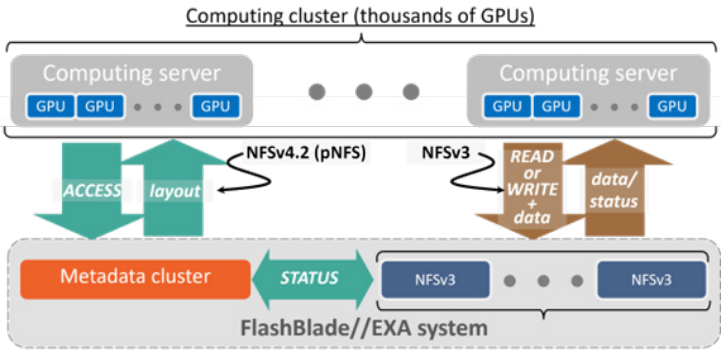

FIGURE 7 Reading and writing existing items

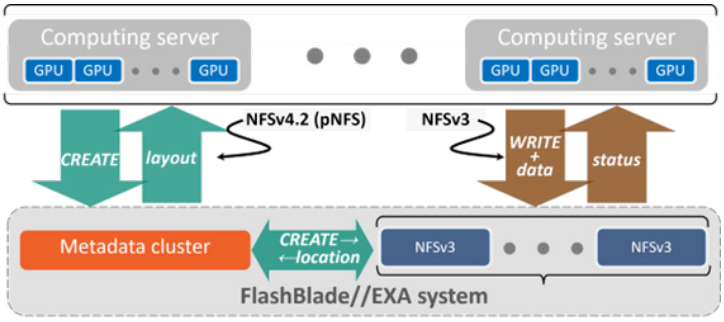

FIGURE 8 Creating new items

Performance

The primary evaluation criteria for AI (and HPC) storage systems are I/O performance and availability. To curate raw data AI workflows may read and write billions of files, but demand for throughput peaks during model training when computing servers must keep GPUs occupied and write checkpoints as often as every few minutes. AI requires both high throughput for reading and writing large items and low latency for metadata access and small item transfers:

Throughput

At publication time, Pure Storage's preliminary testing with publicly available AI benchmarks[5] has already shown that **FlashBlade//EXA** can deliver up to twice the throughput of the best published results, providing upwards of 2.7GB/s of read throughput per GPU in one case. Each data node can deliver upwards of 85GB/s of read throughput. A system with 60 data nodes has been tested at over 5TB/s of read throughput, increasing linearly with the addition of data nodes. The FlashBlade-based metadata cluster does not limit throughput.

Latency

Consistent low-latency metadata access is critical for accessing data items at rates that keep thousands of GPUs at 90%+ efficiency. For example, if a 1,000-GPU training job reads a total of a billion files, reducing OPEN command processing time by a millisecond lowers each GPU's overall processing time by about 17 minutes.

The FlashBlade platform on which the metadata cluster is based is well-known for excellent metadata performance, especially in bulk operations such as changing ownership or access mode of or deleting millions of files.

FlashBlade//EXA systems are designed to deliver excellent performance in both dimensions:

- Because users choose the data nodes and network infrastructure for their systems, they can deploy the data node and network components that provide the most appropriate capacity-performance-cost balance for their environments.
- Purity's unique scale-out design minimizes internal contention and resource locking to deliver up to 20 times the metadata performance of conventional scale-out storage system architectures.

System management

The **Purity//DN** data node software component of **FlashBlade//EXA** includes tools for managing a system's data nodes, including playbooks for configuration, integration, and upgrading. Metadata clusters track data node utilization and status, allocate storage for new data items, and relay alerts from data nodes. At publication time, data nodes are not yet managed through Pure Fusion™, or the Pure1® cloud. "First calls" go to Pure Storage, which may direct users to data server, SSD, or network vendors for further analysis and remediation.

Why FlashBlade//EXA?

AI adopters have options for storage—cloud providers, multi-cloud hybrids, purpose-built in-house systems, and more. Many begin adoption with cloud-based approaches to minimize startup expense and risk and to assess the value of AI to them. But when AI has proven its worth and become integral to an enterprise's IT repertoire, performance, flexibility, security, and cost considerations often motivate a transition to in-house systems.

With AI storage vendors vying for customer attention, what makes **FlashBlade//EXA** systems stand out from the crowd?

Robustness

Metadata is at the core of a disaggregated AI storage system. While many of the curated data items used in AI are reproducible, system metadata must be both durable and highly available. Pure Storage's decade-long track record of "five nines" (or more) of availability over tens of thousands of its systems in production demonstrates that FlashBlade-based metadata clusters are a solid foundation for petabyte-scale AI storage deployments.

Flexibility

FlashBlade//EXA users can configure metadata clusters to meet current requirements and expand them non-disruptively by adding blades and/or chassis as metadata grows. Similarly, they can configure data nodes and networking (or utilize existing equipment) that meet current performance-capacity-cost-availability requirements and constraints and expand as more data and/or throughput become necessary.

Performance

Pure Storage's internal testing has shown that **FlashBlade//EXA** throughput scales linearly as data nodes are added to a system. Users can achieve tens of terabytes per second of throughput by configuring sufficient data nodes and supporting network infrastructure.

The Purity software architecture is unique among scale-out system designs in minimizing internal contention. Metadata performance scales linearly as blades are added to a cluster. The underlying FlashBlade architecture is especially well-known for dramatically improving the performance of bulk metadata operations.

The Pure Storage culture of simplicity

Since its inception, Pure Storage has striven to make storage simple. The company's entry into the complex realm of AI and HPC storage is no exception. From the scale-out metadata cluster concept to flexible data node configurations for which Pure Storage supplies complete Purity//DN operating software and management tools, to parallel file system-specific consulting and support, Pure Storage makes designing, configuring, deploying, and managing storage for AI and HPC as friction-free as it can possibly be.

Looking ahead

Arguably, the complexity of AI systems led to the emergence of cloud-based service providers that allow users to exploit the technology without the expense and "learning curve" of in-house deployments. But as AI becomes part of enterprises' core IT, systems are evolving to become simpler and more cost-effective. Some enterprises are moving their AI in-house to achieve performance, flexibility, security, and cost control that aren't available with cloud services.

Throughout Pure Storage's history, starting with its introduction of affordable enterprise flash storage, through FlashBlade scale-out file and object systems, Evergreen® storage as a service, to today's Enterprise Data Cloud managed by Pure1 and Pure Fusion, the company has consistently identified and met users' pressing storage needs. It introduces basic solutions and enhances them over time to perform better and to be functionally richer and easier to use.

Today, AI's most pressing data needs are reliable high throughput and consistent low latency metadata access for data curation, model development, and production. But AI technology is evolving rapidly, and storage must evolve with it—in performance, functionality, and ease of deployment and use. Pure Storage's solid technology base positions it to lead as AI storage needs evolve over time. For example:

- To support training of larger models, even today **FlashBlade//EXA** metadata clusters can expand non-disruptively to accommodate increasing numbers of data nodes.
- Pure Storage's long experience with highly available storage systems would be a solid foundation for developing highly available data nodes to improve overall system availability.
- FlashBlade enterprise systems support multi-tenancy with quality of service (QoS) guarantees—easily extendable features to accommodate AI and HPC deployments that serve multiple constituencies.
- FlashBlade's mature object storage support is a solid foundation adaptable to metadata for AI applications based on objects rather than files when and if they emerge.
- Pure Storage's DirectFlash technology is a potential foundation for higher-performing, more reliable and cost-effective data nodes than are possible with off-the-shelf SSDs.

The company constantly monitors users' storage requirements and employs its technology to deliver higher performance, more comprehensive solutions as indicated by market demand.

1 | https://en.wikipedia.org/wiki/GPFS

2 | https://en.wikipedia.org/wiki/Lustre_(file_system)

3 | Among many other features unrelated to parallel storage systems.

4 | https://datatracker.ietf.org/doc/html/rfc8435

5 | https://mlcommons.org/benchmarks/storage/. At publication time, Pure Storage's internal testing has been with the Resnet50, 3D U-NET, and Cosmoflow benchmarks. Results have not yet been submitted to MLcommons for certification.

ID2898-01-en-01/26