# A STORAGE ARCHITECTURE FOR AI

## *The Right Storage Can Both Simplify an AI Deployment and Enhance Its Value to the Business*

After years of unfulfilled promise, artificial intelligence (AI) has "arrived." Driven by evolving computing hardware, plummeting costs, and improving software techniques, AI is having major impacts in many fields–finance, medicine, agriculture, security, and more. It is solving real problems whose importance makes cost, reliability, scalability, and ease of use key factors in deployments. Organizations that wish to adopt AI, move a cloud project in-house, or replace a failing infrastructure are often unsure of how to create a durable AI architecture. This brief describes AI's lifecycle data storage needs and shows why Pure Storage's FlashBlade® systems are the perfect storage for an AI project from concept to maturity.

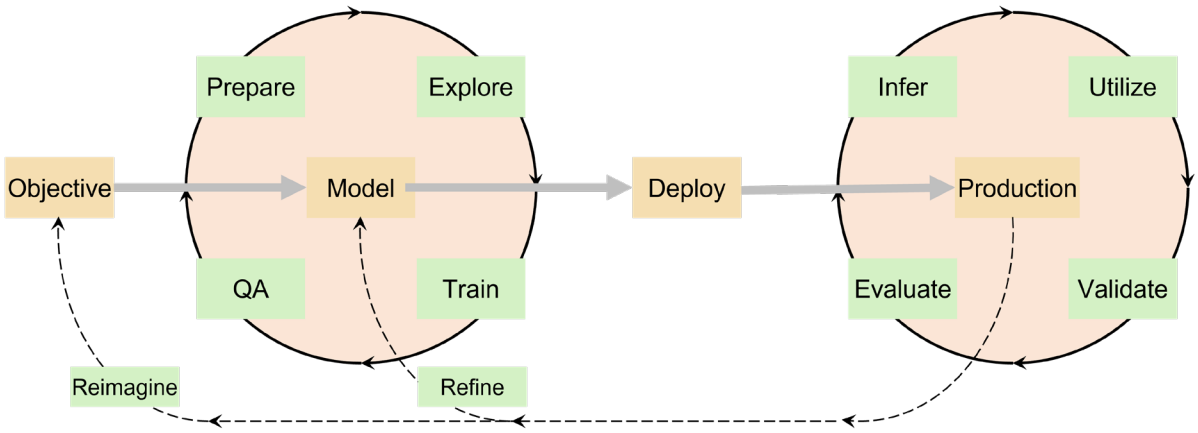



Figure 1: Lifecycle of a Mature AI Project

While every deployment is unique to some degree, the AI project lifecycle illustrated in Figure 1 is common. A project begins with a business objective, e.g.,

- More accurate medical diagnosis
- Better crop yields
- Predictable market fluctuations
- Discovery of computer security breaches
- etc.

AI projects achieve objectives like these by analyzing huge amounts of data that may be readily available or easily acquired. *Data scientists*—usually from a business line rather than an IT organization--develop *models* that use the data to fulfill desired objectives. Projects often seek to minimize startup costs by using public cloud IT services, so organizations sometimes find themselves buying cloud services for several unrelated projects. As models mature, cloud services become expensive and scale poorly, so they shift to in-house production to deliver actual results to the business. These shifts may surprise IT teams who have not been involved in project development, and can be costly and time-consuming, especially if they occur when training data sets already include billions of items. Finally, models evolve as data patterns shift and objectives change. This brief examines storage needs over the life of an AI project:

### MODEL DEVELOPMENT AND TRAINING

AI model development is highly iterative–successive experiments confirm or refute hypotheses. As models develop, data scientists *train* them using sample data sets, sometimes through tens of thousands of iterations. For each iteration, they *augment* data items, slightly randomizing them to avoid *overfitting*–creating a model that is accurate for the training data set but less so with live data. As training progresses, data sets grow, eventually forcing a move from cloud to data scientists' workstations, and ultimately to data center servers that have more extensive computing and storage resources.

### PRODUCTION

When a model produces consistently accurate results, it is put into production. Emphasis shifts from perfecting the model to maintaining a robust IT environment. Production may be interactive–MRI images are usually analyzed immediately–or batch oriented–satellite images acquired over months are analyzed to estimate crop yields or discover soil treatment needs. Whatever the usage mode, production AI must be reliable, deliver timely results, and be easy for non-specialists to use.

### FEEDBACK AND EVOLUTION

Unlike typical legacy applications, AI models do not remain unaltered for extended periods. New data is constantly used to refine them for improved accuracy. Data scientists update training data sets and analyze model outputs, for example to discover pattern shifts that result from changing conditions. They may completely reimagine models because additional inputs become available or because business objectives change. Typical AI deployments evolve continuously throughout their lifetimes.

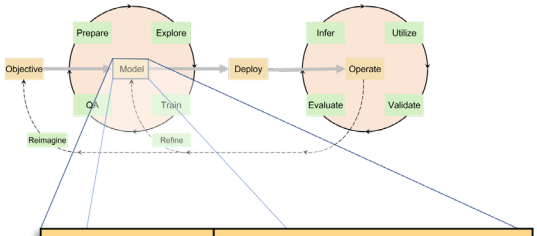
## AI FROM AN IT PERSPECTIVE

AI is computationally intensive, particularly in training and production. It has been the primary motivator for the evolution of graphics processing units (GPUs) into massively parallel computing

engines such as NVIDIA Corporation's A100 series.[1] But sometimes overlooked is the fact that it is also I/O intensive, especially during model development and training. Each stage of an AI project has unique data storage and I/O needs:

## MODEL DEVELOPMENT AND TRAINING

AI is necessarily somewhat speculative. At the beginning of a project, it is often not certain that its objective can be met. Early model development is therefore generally cost-conscious. Experimenters often develop on public cloud services or personal workstations. When a model shows signs of success, however, it quickly outgrows those resources. As projects near production, they need high-performance computing and terabytes of data. Workstations are inadequate and public clouds are often unaffordable.




| Modeling Stage | Storage Needs |
|---|---|
| Concept | Individual (e.g., workstation, cloud) |
| Preprocessing | Random-Sequential flexibility |
| Development | Scalable multi-job performance |
| Training | Rapid growth, fast enumeration |

Figure 2: Storage Requirements During Modeling

The data used to develop AI models often has multiple sources–event logs, transaction records, images, IoT inputs, and so forth. It usually requires preprocessing and *balancing* so that no one input dominates. Different streams must be reconciled, for example, by adjusting measurement units or correlating time lines. During this stage, items are often *annotated* or labeled by linking them to small files or objects.

**Storage requirement**: *Model development jobs typically read raw data randomly and write preprocessed items sequentially. Storage systems should provide low latency for small random reads concurrently with high sequential write throughput. "Tuning" storage systems for specific access modes is usually counterproductive.*

As development progresses, data sets grow, and cooperating data scientists access them concurrently, augmenting items dynamically in thousands of variations to avoid overfitting.

**Storage requirement**: *Storage expansion starts to become important at this stage, but scalable performance as increasing numbers of concurrent jobs access data is the real key to success. Data sharing by workstations and servers and rapid, non-disruptive capacity expansion are the most important storage attributes.*

[1] https://www.nvidia.com/en-us/data-center/a100/
Pure Storage and NVIDIA have jointly developed the AIRI//S system architecture based on DGX servers with A100 GPUs and FlashBlade//S storage. See https://www.purestorage.com/products/file-and-object/flashblade-s.html.

During training, data set sizes increase manyfold, often to petabytes. Each training job typically reads data items randomly. The overall training process consists of many concurrent jobs that access the same input data sets. Multiple jobs competing for data access intensify the overall random I/O load at this stage.

> ***Storage requirement**: The transition from model development to training needs storage that can (a) expand non-disruptively to hold billions of data items, and (b) can provide fast multi-host random access for concurrent training jobs. Also at this stage, storage system "tuning" for specific I/O patterns is generally counterproductive.*

Training jobs often *decompress* input data (e.g., convert compressed images to bitmaps), and augment or *perturb* it and randomize input order, again to avoid overfitting. Randomization requires data item *enumeration*–querying storage to obtain electronic lists of the billions of items in a training data set.

> ***Storage requirement**: Enumerating billions of training data items serially is much too time consuming to be practical. Parallel enumeration is an absolute necessity.*

Training jobs may run for days, so most write periodic checkpoints so they can restart after a failure. Training workloads dominated by random reads are therefore intermixed with large sequential writes for checkpointing.

> ***Storage requirement**: Storage systems should be capable of sustaining the intensive random access that concurrent training jobs require, even during occasional bursts of large sequential writes as jobs are checkpointed.*

### DEPLOYMENT AND OPERATION

Production AI has a lot in common with legacy applications. Service availability, data asset protection, seamless scaling of capacity and performance, and operational simplicity as IT teams' personnel changes become key considerations.

> **Storage requirement**: *AI storage should be capable of "24x365" operation throughout a project's life, self-healing when components fail, and non-disruptive to expand and upgrade. It should protect against human error (e.g., with data set snapshots), and fit easily into data center "ecosystems."*

### FEEDBACK AND EVOLUTION

Model refinement is essentially periodic retraining with data sets that include new data and omit items that are no longer relevant. Production data should be readily accessible to data scientists for training updates and for model review and reexamination.

**Storage requirement**: *Data scientists' need for production data to adjust models and explore changing patterns and objectives argues strongly for a consolidated platform–a single storage system that meets the needs of all project phases and allows development, training, and production to easily access dynamically evolving data.*

## AI STORAGE PITFALLS TO AVOID

These requirements suggest what to avoid in selecting storage for an AI project:

### CONFIGURATION RIGIDITY

It is difficult to predict lifetime storage needs early in a project. Storage that is awkward or impossible to expand or upgrade requires time and resources to reconfigure, or worse, complete replacement. Even upgradable storage can be problematic if upgrading causes service outages, especially during production, when results drive important decisions.

### DATA ISOLATION

Data that can't easily be shared among data scientists or between data scientists and production must be copied from where it is to where it's needed. Model development stops during copying. As data sets grow, copies can take hours to complete, and moving data between development and production equipment is disruptive. Moreover, in addition to being time and resource-consuming, copying is prone to human error.

### INFLEXIBILITY

I/O requirements vary throughout the life of an AI project. Storage that is highly optimized for narrow use cases (e.g., by data set locations, block sizes, file types, etc.) can be difficult to adjust to changing conditions such as when adding new input types to a model. "Tuning" requires expertise, and even experts don't always get it right.

### LACK OF ROBUSTNESS

If an AI project is important enough to invest in, it follows that it is important enough to keep available. Data scientist productivity during model development and training is important, but in production, the ability to deliver results when they're needed is vital. Storage that can't survive component failures, be repaired or upgraded while it's online, or recover from user and administrative errors isn't up to the job and shouldn't be considered.

## FLASHBLADE//S®:
## WHAT IF A STORAGE SYSTEM WERE DESIGNED SPECIFICALLY FOR AI?

The Pure Storage® FlashBlade//S *Unified Fast File and Object* (UFFO) system sets new standards for performance, scalability, and simplicity in high-capacity file and object data storage, making it perfect for AI. The all-flash blade-based systems integrate hardware, software, and networking to offer higher storage density with lower power consumption and heat generation than other systems, along with versatile performance to support virtually any combination of file and object workloads. FlashBlade//S systems range from 168TB to 19.2PB of physical capacity (about 180TB to 25PB of *usable* storage[2]), and are available in either capacity-optimized or performance-optimized configurations.

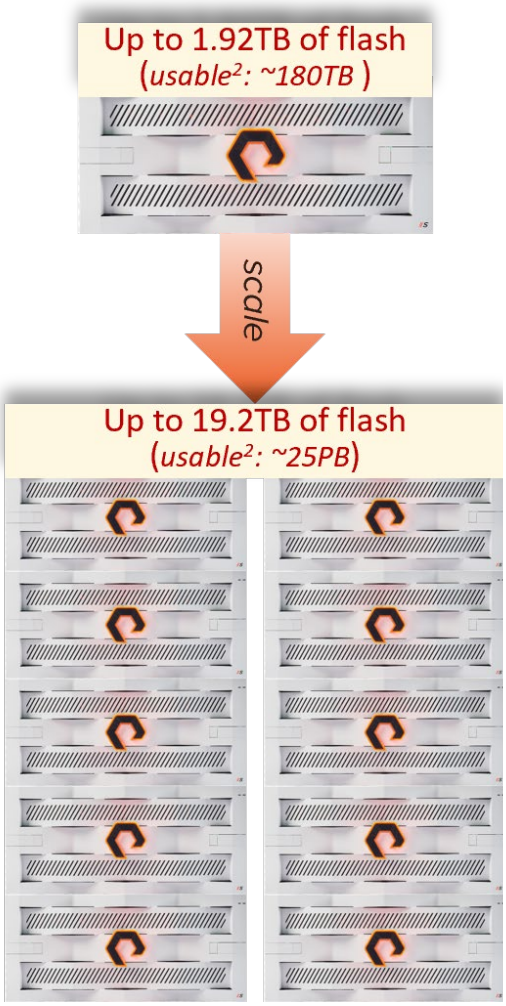



Figure 3: FlashBlade's Capacity Range

FlashBlade//S is a symmetric scale-out hardware platform that uses Purity//FB operating software to host file systems and object stores that meet the storage and I/O needs of AI workloads as they evolve during project lifetimes. The appendix briefly describes the system's hardware and software.

Purity//FB is a distributed symmetric operating system–all blades in a system run identical software. Both file (NFS and SMB) and object (S3) protocols access data through a common back-end "engine" that automatically balances capacity utilization and I/O across all blades in a system regardless of client workload. As a result, systems constantly "tune" themselves. There is no need for administrators to relocate data sets, adjust operating parameters, or shut down for expansions and upgrades, as projects progress from model development, through I/O-intensive training, to full production.

[2] Usable capacity is that available for user data, net of RAID overhead and system metadata. The 180TB and 25PB usable capacity estimates are based on 2:1 data compression. Compressibility varies based on the nature of data. For example, text and tabular data usually compress 2:1 or more, whereas images, streams, and encrypted data are essentially incompressible.

FlashBlade//S scalable capacity, versatile performance, and ease of administration can help AI projects run smoothly throughout their lifetimes. The system's integrated networking connects individual data scientists to their data, delivers high performance access for I/O-intensive training jobs, and provides the reliability that production usage requires.

## FLASHBLADE//S AND AI STORAGE REQUIREMENTS

### CAPACITY (INITIAL AND EXPANSION)

The minimum FlashBlade//S system is a chassis containing seven blades, for a physical capacity of about 168TB; the maximum 10-chassis system contains 19.2PB of physical QLC flash. Capacity expansion is non-disruptive, and there is no need for data migration.

*An AI project that uses a single-chassis system during early model development can expand the system as data requirements grow during training, and continue to expand as more live data is accumulated during production.*

### DATA SHARING

Each FlashBlade//S chassis contains redundant *fabric I/O modules* (FIOMs) that interconnect its blades and provide eight[3] 100Gbps-capable external ports. Single-chassis systems' FIOMs connect directly to the client network; in multi-chassis systems they connect to *external fabric modules* (XFMs). Administrators have complete control over user and application access to files and object stores.

*With 8 x 100GbE ports per chassis, there is adequate bandwidth for data scientists to experiment, even as training jobs impose heavy I/O loads on data and deployed models perform production tasks.*

### PERFORMANCE WITH VARYING I/O LOADS

As an all-flash system, FlashBlade//S doesn't use mechanical motion to access data. Every file and object is accessible in approximately the same amount of time. Time-to-first-byte access is the same for 1KB item labels as for 50MB images. Systems are not tuned to specific block sizes, file types, object sizes, or data layouts.

*Because it treats all I/O requests equally, FlashBlade//S can support data scientists developing models, training jobs' intensive random I/O loads and checkpoint writes, and production I/O in a single system. As maturing projects require system expansion, FlashBlade//S automatically adapts data placement and I/O distribution to utilize all available resources effectively.*

[3] Four external ports per blade are supported in the initial product release.

## PARALLEL FILE OPERATIONS

A *RapidFile Toolkit for Linux*,[4] available to FlashBlade//S users at no incremental cost, takes advantage of Purity//FB parallel operation to greatly accelerate several operations commonly used in AI projects, such as bulk copying, enumeration, changing properties, etc. With large numbers of files, RapidFile tools can perform tasks in minutes that typically take hours using conventional operating system tools such as **`ls`** and **`chown`**.

*Creating copies, changing ownership, randomizing billions of files to prevent overfitting, and so forth help speed up AI model development. With RapidFile tools, FlashBlade//S performs these and similar tasks easily and quickly.*

## DATA RELIABILITY

FlashBlade reliability has been field-proven by thousands of installed systems. Purity//FB protects against data loss due to DFM and blade failures, and when a component does fail, automatically initiates distributed rebuilds to restore full protection as quickly as possible.

*It's important to keep data scientists productive by providing them with reliable storage. But when a project moves into production and business decisions are made based on the results it produces, the data itself becomes vital, as does reliable access to it.*

## EASE OF USE

Simplicity and ease of use have been Pure Storage hallmarks since the company's earliest days. FlashBlade//S exemplifies this. Internal connections are via chassis midplanes, minimizing cabling and simplifying expansion. The software-defined networking is simple to set up and administer. Administrators use CLI or GUI interfaces to create file systems and object buckets by specifying names and size limits; placement, allocation, and dynamic tuning are all automatic. REST APIs help integrate FlashBlade//S administration with data center automation tools. Pure1® provides centralized monitoring for all of an organization's systems, and enables remote troubleshooting and upgrading by Pure's customer support teams.

*Because storage requirements vary greatly during different AI project stages, FlashBlade//S non-disruptive expansion and upgrading keep workflows functioning smoothly. Administrators can repurpose capacity with a few keystrokes. Simplicity minimizes training requirements as data center operations personnel come and go. Overall, FlashBlade//S administration is one of the simplest tasks in a production data center.*

[4] https://support.purestorage.com/Solutions/Linux/Linux_Reference/RapidFileToolkit

## AT THE END OF THE DAY...

This brief demonstrates that FlashBlade//S is an ideal storage system for an AI project's entire lifecycle. It relieves both data scientists and production IT teams from most common storage and networking concerns, allowing them to focus on modeling and providing reliable, high-performing, easily expandable storage as projects progress from development to production to further enhancement.

Available in performance-optimized and capacity-optimized versions that scale on demand, FlashBlade//S systems whose primary mission is storage for an AI project can often do double duty by hosting data for other applications such as data lifecycle management (e.g., Kafka), monitoring and alerting for the data center, and so forth.

Organizations can use FlashBlade//S as a storage platform for healthcare, genomics, mineral exploration, financial, and many other AI applications, and share capacity and I/O bandwidth with data-intensive applications in areas like analytics, database backup, software development, media and entertainment post-production, electronic design automation (EDA) and so forth.

## FOR FURTHER INFORMATION...

Additional information on topics discussed in this brief is available at:

**FlashBlade//S technology**
https://www.purestorage.com/products/file-and-object/flashblade-s.html

**AIRI//S (AI-Ready Infrastructure)**
https://www.purestorage.com/products/ai-infrastructure/airi-s.html

**The RapidFile Toolkit**
https://support.purestorage.com/Solutions/Linux/Linux_Reference/RapidFileToolkit

# APPENDIX: INSIDE FLASHBLADE//S

## UNIFIED STORAGE FOR COMPLEX PROJECTS

The FlashBlade//S hardware architecture is *scale-out*–storage capacity and performance both increase with the addition of blades. Paired with Purity//FB software, systems are high-performing file and object stores suitable for virtually any workload. The FlashBlade//S models described in this appendix support physical flash capacities from 168TB in a minimally configured single chassis to nearly 20PB in the highest-capacity 10-chassis system. Technical briefs TB-210201 and TB-220601, available at https://support.purestorage.com, describe the FlashBlade//S hardware architecture and Purity//FB software respectively in greater detail.

## HARDWARE

A FlashBlade//S chassis (illustrated in Figure 4) is a self-contained, rack mountable storage system for files and/or objects that provides fast access to hundreds of client systems. A 5RU (8.75 inch high) chassis holds between 168TB and 1,920TB of QLC flash. With always-on data compression, a fully populated capacity-optimized chassis can store over 2PB of user data.[5] A maximally configured 10-chassis capacity-optimized system contains 19.2 petabytes of flash and can hold about 25PB of compressible user data.

Figure 4: FlashBlade//S Chassis

### BLADES

As its name implies, FlashBlade//S is based on hot-swappable compute/storage *blades* connected to a chassis midplane that distributes power and hosts two fabric I/O modules (FIOMs) that interconnect the blades and provide access to the client network.

Each blade consists of a mainboard containing a processor, DRAM, and dual NICs, and hosts between one and four *DirectFlash® modules* (DFMs) connected via on-board PCIe. FlashBlade//S can be configured with *capacity-optimized* blades that use extensive compression to minimize data's footprint, or with *performance-optimized* blades that streamline data paths for maximum throughput.


Figure 5: Blades and DFMs

[5] Net of RAID overhead and metadata. Assumes 2:1 data compression, typical for text and tabular data.

A chassis contains 10 blade mounting slots; a minimum system contains seven blades. The minimum unit of expansion is a single blade with the same number and type of DFMs as the other blades in the chassis.

### DIRECTFLASH MODULES

In FlashBlade//S, 24TB (performance-optimized) and 48TB (capacity-optimized) DFMs use QLC flash to provide persistent storage and high-speed NVRAM in which data is *staged* prior to writing. Staging makes write performance consistently fast. Specialized hardware transfers data between blade DRAM and DFMs, calculates flash page ECCs, and encrypts all data and metadata. Unlike SSDs, DFMs have no flash translation layer (FTL)—blades address on-board flash directly which optimizes performance and media longevity. DFMs also support *scatter-writing* to non-contiguous flash locations, a key requirement for atomic operations that require access to DFM partitions controlled by two or more blades.

### NETWORKING

Dual FIOMs in the FlashBlade//S chassis interconnect blades and connect to the external network for client access. The FIOMs contain state-of-the-art Ethernet switches with a total of 2Tbps cross-sectional bandwidth and eight external ports,[6] each capable of 10, 25, 40, or 100Gbps operation.

In single-chassis systems, FIOMs connect to the data center network for client access; in multi-chassis systems, chassis connect to each other and to the data center network via *eXternal Fabric Modules* (XFMs) that contain switches similar to those in the FIOMs.

## SOFTWARE

Purity//FB is specifically designed to manage flash storage. The software is symmetric—each blade's compute module runs the three types of processes depicted in Figure 6:

### ENDPOINTS

Route client I/O requests received from FIOMs to the authorities responsible for executing them.

### AUTHORITIES

Execute client requests by reading and writing data locally or on other blades. Each authority controls a separate *partition* of the flash and NVRAM on every blade.

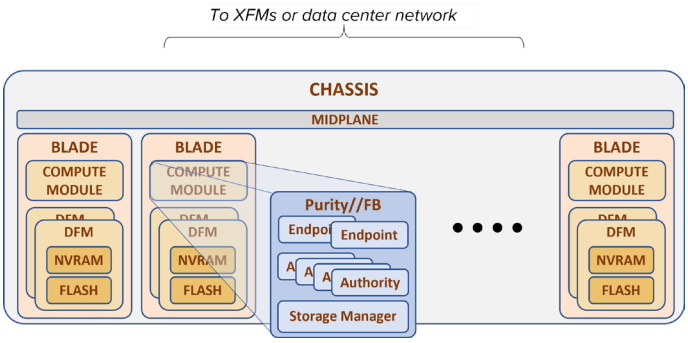



Figure 6: Purity//FB Structure

[6] Four external ports per chassis are enabled, and intra-chassis communication is at 50Gbps in initial FlashBlade//S systems.

### STORAGE MANAGERS

Execute read and write requests for data stored on the blade's DFMs. Requests may come from the local blade or may be received from other blades via the midplane.

## LOAD BALANCING

Purity//FB is distributed by design. FIOMs distribute connection requests randomly to Endpoints to balance load regardless of client access pattern. Endpoints route requests to authorities based on the data it addresses. Each authority controls its own flash and NVRAM partitions on every blade, thus largely eliminating the need for locking.[7] They stage incoming data (3x-mirrored) in NVRAM partitions and pack it densely in large buffers; no space is wasted by "rounding" to fixed block sizes. As buffers fill, authorities pass them to the local Storage Manager to be RAID-protected and written to flash.

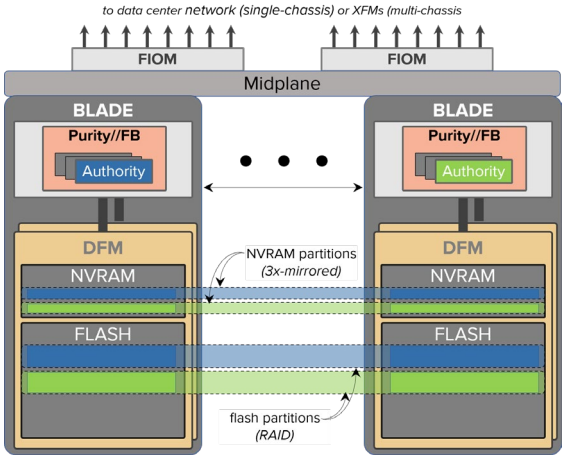



Figure 7: NVRAM and Flash Partitions

Authorities also perform background functions, including flushing write buffers when they fill, managing the contents of their NVRAM partitions, *garbage collecting* (reclaiming space occupied by overwritten data), and *scrubbing* media to detect and correct latent read errors. The back-end I/O load from these functions is inherently balanced, because each authority performs them on separate flash and NVRAM partitions.

Authorities are *stateless*, so they can run on any blade. When a system is expanded or when a blade fails, some authorities move to different blades, but continue to control their original flash and NVRAM partitions from their new locations.

*With designed-in load balancing, FlashBlade//S systems automatically optimize themselves dynamically. No administrative effort is required (or indeed, possible) for load balancing. Self-tuning is what makes FlashBlade//S able to serve the needs of data scientists, model training, and AI production simultaneously.*

[7] For complex requests, such as directory removals involving multiple blades, authorities provide atomicity by scatter-writing updates to multiple NVRAM partitions.

## FILE SYSTEMS AND OBJECT STORES

Purity//FB authorities cooperate to implement a highly scalable distributed back-end store upon which both NFS and SMB file systems and S3 object stores are directly based. There is no "layering" of data access protocols.

*Clients can access any file system's contents using either of the NFS or SMB protocols–a must for data sets used both by data scientists' individual workstations and Linux servers performing training and production tasks. The system supports the widely used S3 protocol for object stores, so users have flexibility in choosing AI development and processing tools.*

## FAULT PROTECTION AND RECOVERY

Like all Pure Storage products, FlashBlade//S is designed for a lifetime of continuous operation. Purity//FB protects data against DFM and blade failures with a type of RAID designed specifically for flash. Systems can withstand simultaneous failure of any two DFMs or blades, even in the same chassis, without loss of data or client access to it. If a DFM or blade fails, the software automatically initiates distributed rebuild procedures to restore full data protection as quickly as possible.

# FlashBlade//S: bringing the modern data experience to AI

*Simplifying and accelerating AI workflows at any scale*