PURE STORAGE®
Uncomplicate Data Storage, Forever

BUSINESS WHITE PAPER

FlashBlade//S: Storage Built for AI

The right storage platform can both simplify an AI deployment and enhance its value to your business.

Contents

Introduction

Driven by the availability of high performance computing hardware, AI software evolution, and plummeting cost, artificial intelligence (AI) has "arrived," with broad and accelerating adoption among both governments and private sector enterprises by solving complex, real-world problems. The rising importance of AI now makes cost, reliability, scalability, and ease of use key factors in deployments. Yet organizations that want 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 white paper describes data storage needs over a typical AI lifecycle and shows why Pure Storage® FlashBlade® systems meet AI storage needs from concept to maturity.

While every deployment is unique to some degree, the AI project lifecycle illustrated in Figure 1 is common.

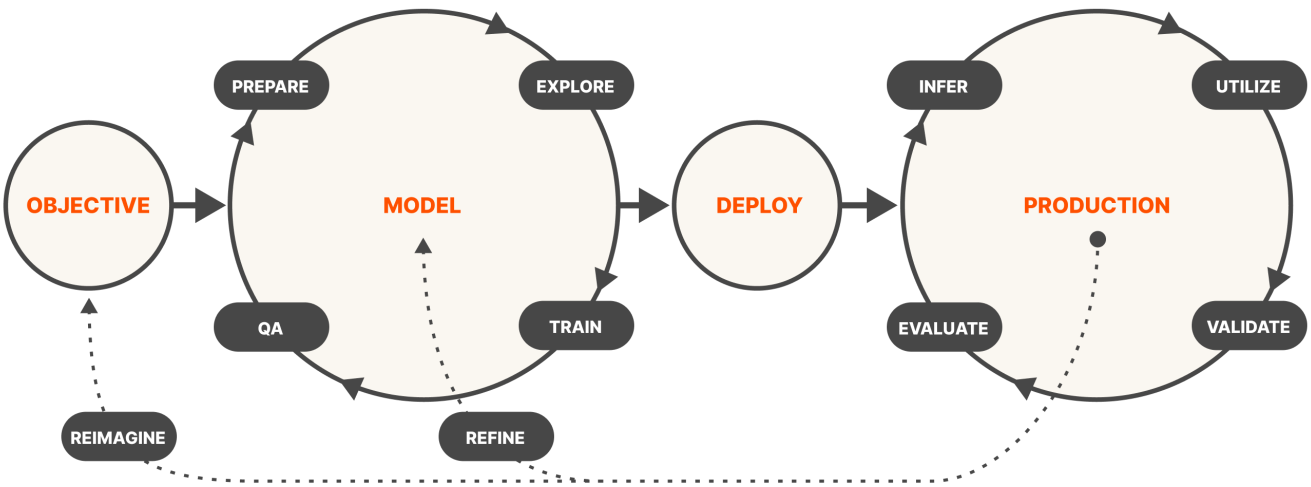


Figure 1. Lifecycle of a mature AI project

AI projects usually begin with a specific business objective, such as:

- More accurate medical diagnoses
- Better crop yields
- Predictable market fluctuations
- Discovery of computer security breaches

Data scientists—usually from a business line rather than an IT organization—develop *models* that analyze huge amounts of 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 and data sets grow, cloud services become expensive and scale poorly so teams shift to in-house production to deliver actual results to the business. These shifts can surprise IT teams that 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 paper 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 with 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 or batch oriented. For example, MRI images are usually analyzed immediately, whereas satellite images are typically analyzed in batches acquired over weeks or months to estimate crop yields of 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 and H100 systems.[1] But sometimes overlooked is 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.

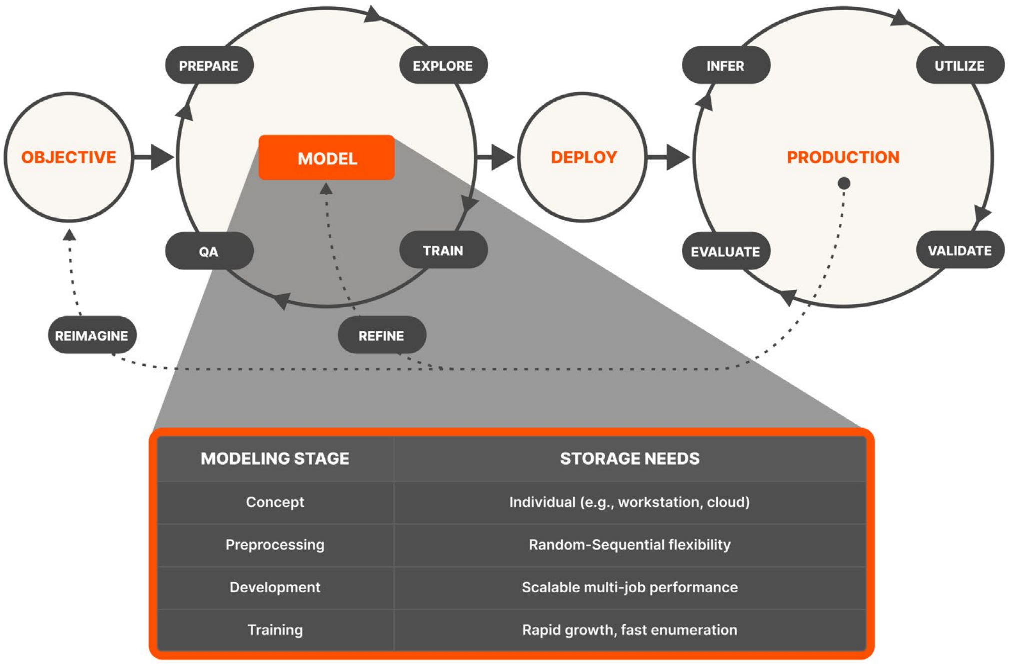
Model Development and Training

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 models on public cloud services or personal workstations. But when a model shows signs of success, it quickly outgrows those resources. As projects near production, they need high-performance computing and storage for many terabytes or even petabytes of data. Workstation resources are inadequate, and public cloud data access charges can make them unaffordable.

The data used to develop AI models often originates from multiple sources such as event logs, transaction records, images, and IoT inputs. 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 timelines. During this stage, items are often *annotated* or labeled by linking them to small files or objects.

---

[1] https://www.nvidia.com/en-us/data-center/dgx-h100/
Pure Storage and NVIDIA have jointly developed the AIRI//S® system architecture based on DGX systems with FlashBlade//S storage. See https://www.purestorage.com/products/file-and-object/flashblade-s.html.



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

> STORAGE REQUIREMENT: Model development jobs typically read raw data randomly and write preprocessed items sequentially. Storage systems should be capable of providing 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 from different workstations concurrently, augmenting items dynamically in thousands of variations to avoid overfitting.

> STORAGE REQUIREMENT: Storage expandability 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.

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 both expand non-disruptively to hold billions of data items, and 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 enterprise business applications. Service availability, data asset protection, seamless scaling of capacity and performance, and operational simplicity as IT team 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

The above 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. 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.

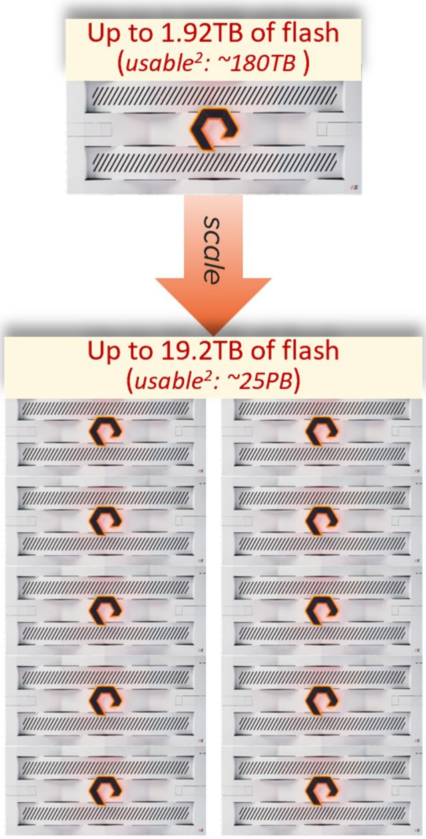


Figure 3: FlashBlade//S capacity range

FlashBlade//S is a symmetric scale-out hardware platform that uses Purity//FB operating software to host file systems and an Object Store that meet the storage and I/O needs of AI workloads as they evolve during project lifetimes.

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 of a system's blades 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.

The scalable capacity, versatile performance, and ease of administration from FlashBlade//S 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, all 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.

- **Parallel file operations:** A RapidFile Toolkit for Linux,[4] available to all FlashBlade 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 when done 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 toolkit, FlashBlade//S performs these and similar tasks easily and quickly.

---

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

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

- **Data reliability:** FlashBlade reliability has been field-proven by thousands of installed systems. Purity//FB protects against data loss due to DirectFlash® module (DFM) and blade failures, and when a component does fail, it 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, along with 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 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.

At The End of The Day...

This paper 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 they can 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 more.

Additional Resources

Next Steps

Customers can request the following technical reports from their Pure Representative or log into https://support.purestorage.com/.

- Future-Proofing FlashBlade (TB-210201)
- The Purity//FB Architecture (TB 220601)

Further Reading

- FlashBlade//S technology
- AIRI//S (AI-Ready Infrastructure)
- RapidFile Toolkit

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041

PS2324-01 10/2022