PEAK:AIO Open pNFS

Parallel AI Storage Architecture for High-Performance, GPU-Accelerated Workloads

A technical whitepaper examining how PEAK:AIO Open pNFS eliminates the fundamental storage bottlenecks that throttle modern GPU clusters — delivering true parallel data access, linear scalability, and NVMe-native performance across AI training, inference, analytics, and HPC workloads.

TECHNICAL WHITEPAPER | NFSV4.2 PNFS ARCHITECTURE | GPU-OPTIMIZED STORAGE

Executive Summary

Artificial Intelligence (AI), Machine Learning (ML), and High-Performance Computing (HPC) workloads are fundamentally reshaping enterprise infrastructure requirements. GPU clusters continue to scale in both size and raw compute throughput, yet storage architectures remain a persistent and costly primary bottleneck. Traditional NAS systems introduce metadata serialization delays, network chokepoints, and inefficient single-path data flows that systematically limit GPU utilization and extend time-to-insight — directly undermining infrastructure ROI.

PEAK:AIO Open pNFS eliminates these structural constraints through a fully parallelized, standards-based architecture built on Parallel NFS (pNFS), as defined by the NFSv4.2 specification. By enabling direct, parallel client-to-storage data paths — bypassing centralized controller congestion — PEAK:AIO delivers ultra-high aggregate throughput, predictably low-latency access, and linear scalability across the full spectrum of AI and HPC workload profiles.

Unlike legacy scale-up NAS or proprietary parallel file systems that impose vendor lock-in and operational complexity, PEAK:AIO takes a differentiated approach that combines open standards compliance with advanced engineering optimizations purpose-built for GPU-scale computing.

True Parallel Access

pNFS-native architecture enables simultaneous client-to-storage I/O across all nodes, eliminating controller serialization entirely.

NVMe-Optimized

PCIe Gen4/Gen5 NVMe storage tiers deliver sub-millisecond latency and multi-GB/s per-node sustained throughput for demanding workloads.

AI-Driven Intelligence

Real-time telemetry, workload profiling, and predictive analytics continuously optimize storage behavior for GPU workload demands.

Ethernet-Native

Full performance delivered over high-speed Ethernet (100/200/400GbE), with no InfiniBand requirement and no proprietary client stack.

The result is a GPU-optimized storage platform capable of sustaining multi-node, multi-GPU workloads at scale while preserving the operational simplicity and standards compliance that enterprise infrastructure teams require.

Section 1: The AI Storage Challenge

1.1 GPU Acceleration Outpaces Storage

The modern AI infrastructure stack is defined by an ever-widening performance asymmetry. NVIDIA H200/B200/B300 GPU clusters, operating within distributed deep learning frameworks such as PyTorch, TensorFlow, and Horovod, generate I/O demands that traditional storage architectures were never designed to accommodate. Training runs now routinely operate against multi-petabyte datasets, require high-frequency checkpoint writes to preserve model state, and depend on highly parallel I/O patterns that stress every layer of the storage stack simultaneously.

While GPU compute performance has scaled exponentially — following a trajectory driven by advances in silicon, interconnect bandwidth, and framework optimization — storage throughput and access latency have not kept pace. The consequence is measurable and economically significant: GPU nodes idle while waiting on data, a condition sometimes described as "GPU starvation." In large-scale training clusters, even modest storage-induced delays compound across hundreds or thousands of GPU-hours, translating directly into wasted capital expenditure and extended time-to-model convergence.

The Four Core I/O Demands of AI Workloads

Large sequential reads: Streaming multi-GB mini-batches from storage to GPU memory during training iterations at sustained high bandwidth

High parallelism: Dozens to hundreds of GPU workers simultaneously requesting data, requiring non-blocking, concurrent I/O paths

Frequent checkpoint writes: Periodic serialization of model weights and optimizer state at intervals ranging from minutes to hours, demanding high write throughput

Mixed file-size access: Simultaneous handling of large tensor files and small metadata objects, requiring a storage tier optimized across both access profiles

Storage-Induced Bottlenecks

When storage systems fail to match these demands, the failure manifests across four distinct dimensions:

Metadata Bottlenecks

Centralized namespace operations serialize file open, stat, and close calls — limiting effective concurrency

Controller Congestion

Single-controller NAS designs concentrate all data path traffic through one hardware chokepoint

Network Hot Spots

Centralized data routing creates uplink saturation on specific switch ports, throttling aggregate bandwidth

Unbalanced Access

Without intelligent data distribution, some storage nodes become heavily loaded while others remain underutilized

1.2 Limitations of Traditional NAS

Conventional NAS architectures, including both legacy scale-up and many first-generation scale-out designs, share a fundamental structural limitation: they route all I/O through centralized controller hardware. This design choice, inherited from an era when storage access was predominantly sequential, client-light, and latency-tolerant, creates systematic performance ceilings that cannot be engineered away through incremental hardware upgrades.

In a traditional NAS model, every read and write operation — regardless of file size, access pattern, or client count — traverses the same controller logic, metadata processing stack, and network uplinks. As the number of GPU clients scales from tens to hundreds, contention for these shared resources intensifies.

Metadata operations, which in AI workloads occur at extremely high frequency due to dataset sharding, checkpoint management, and framework-level file management, serialize through the metadata server, creating latency spikes that cascade across all connected clients simultaneously.

Scale-out NAS architectures partially address throughput limitations by distributing data across multiple nodes, but they frequently retain metadata bottlenecks by continuing to centralize namespace operations on dedicated metadata servers without pNFS-style layout offloading. Furthermore, many scale-out implementations depend on proprietary protocols or client agents, introducing operational complexity, version dependency, and vendor lock-in that conflict with enterprise standardization objectives.

1

Client Requests Data

GPU worker node issues NFS read/write to centralized controller

2

Controller Serializes

All I/O processed through single metadata + data path — creating a queue

3

Network Chokepoint

Traffic concentrated on controller uplinks — bandwidth ceiling reached

4

GPU Starvation

GPU compute units idle, waiting for data — infrastructure ROI collapses

For AI training workloads characterized by large sequential reads, extreme parallelism, frequent checkpoint writes, and mixed small-file/large-file access, traditional NAS architectures fail to deliver consistent, sustainable performance at scale. The architectural mismatch is fundamental — not addressable through software tuning, caching layers, or incremental hardware refresh cycles.

Section 2: Introducing PEAK:AIO Open pNFS

PEAK:AIO Open pNFS is a parallel file access architecture designed from first principles to address the specific demands of AI and HPC workloads. Rather than adapting a general-purpose NAS platform with AI-specific tuning overlays, PEAK:AIO was architected around the NFSv4.2 pNFS standard — a protocol extension explicitly designed to decouple metadata operations from data I/O, enabling clients to communicate directly with distributed storage targets in parallel.

The pNFS extension to NFSv4.2, defined in RFC 5661 and RFC 5664, introduces the concept of storage layouts — client-side maps that describe the physical or logical location of file data across one or more storage devices. When a client opens a file on a pNFS-capable system, the Metadata Server (MDS) returns a layout that instructs the client precisely where data segments are stored and on which Data Servers (DS) they reside. The client then performs I/O directly to those data servers — bypassing the MDS entirely for data-plane operations — and commits or returns the layout on close.

This architectural separation of the metadata path and the data path is the foundational mechanism that enables PEAK:AIO to deliver parallelism, scalability, and performance that are structurally impossible in centralized NAS designs. The MDS processes control-plane metadata requests — namespace queries, layout allocations, lock management — while data servers handle the full volume of I/O traffic in parallel. As data servers are added to the cluster, aggregate data throughput scales linearly, with no corresponding increase in MDS load for data operations.

True Parallelism

Eliminate controller bottlenecks by enabling simultaneous direct client-to-storage I/O across all data nodes in the cluster

Linear Scalability

Performance scales predictably with each added data node — no architectural limits impose performance ceilings

Standards-Based Access

Full NFSv4.2 compliance ensures broad client compatibility without proprietary agents or vendor-specific kernel modules

NVMe-Native Performance

Storage architecture built around NVMe SSDs with PCIe Gen4/Gen5 — delivering sub-millisecond latency at scale

AI-Aware Telemetry

Real-time workload profiling and automated optimization driven by a dedicated AI Storage Intelligence Engine

Section 3: pNFS Architectural Overview

3.1 PEAK:AIO pNFS Architecture

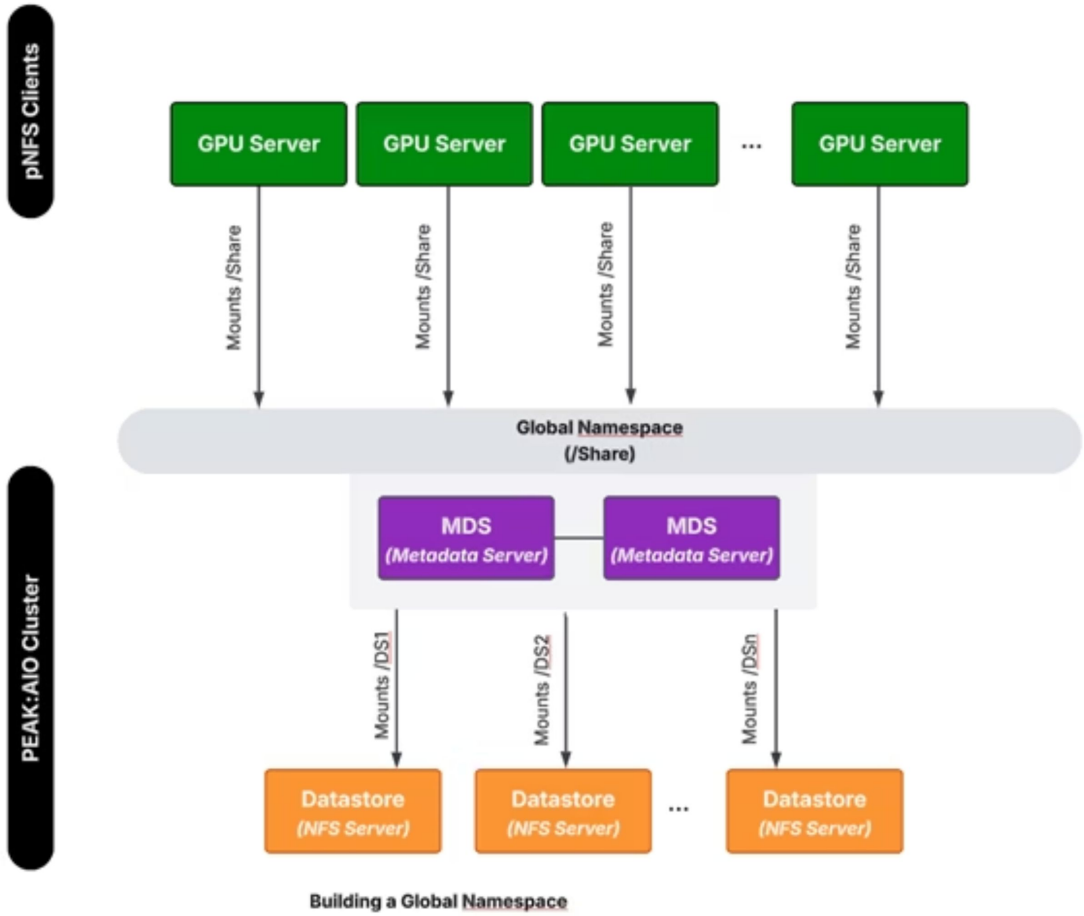
The PEAK:AIO pNFS cluster is designed to provide a global namespace over a shared set of nodes in the cluster, which can be scaled in performance and capacity by adding more nodes. Clients, through the pNFS protocol, will have access to these added resources (automatically and non-disruptively) and will continue to see all the data in the cluster as a single namespace. The goal of the following sections is to detail the elements in the cluster and how they work together to achieve these behaviors.

3.2 pNFS Cluster/Servers

The design goal of pNFS is to provide 2 services to clients: metadata operations and data operations. By separating these services as different node types, a pNFS cluster can accelerate and scale both types of operations better than traditional clusters.

Metadata server nodes (MDS) provide low-latency responses to metadata operations such as retrieving a file's name, permissions, and location (on the datastores).
Datastore nodes (DS) store the file contents and provide high-bandwidth paths to NFS clients for read and write operations.

A typical cluster starts with 2 metadata server nodes (MDS) and several datastores (DS). Each DS is set up as an NFS server, exporting its storage as a single share. The MDS pair will then mount these shares and re-export them as one large share, building a global namespace for the client to use. As data is added to the cluster, the MDS nodes will ensure that data is balanced across all of the DS nodes. No space management is required for individual nodes.



Building a Global Namespace

3.3 pNFS Clients

There is no special software needed by the clients to use the cluster. The pNFS client (version 4.2 with FlexFile support) is available in all modern Linux distributions. For clients, such as GPU or CPU compute nodes, to use the cluster, they simply mount the cluster's share using standard NFS mount commands, specifying a few mount options (e.g., rdma, nconnect, version) mainly for optimization purposes.

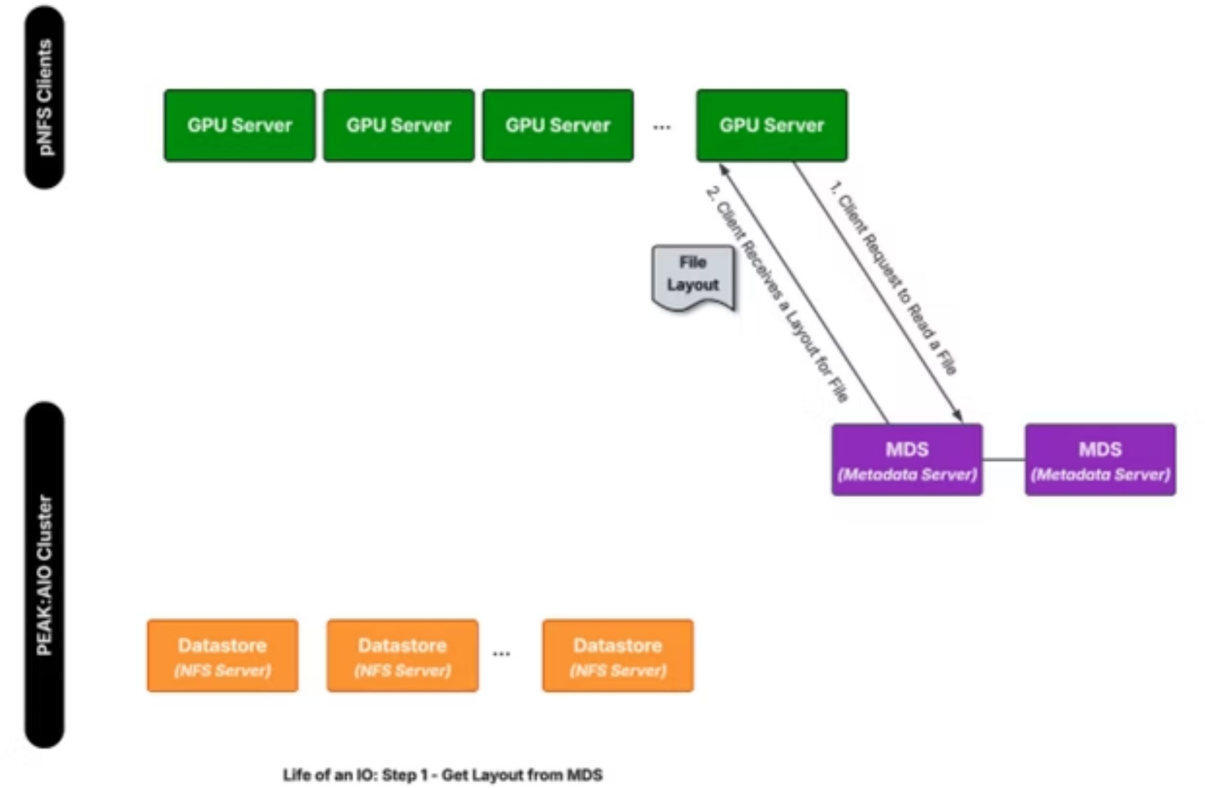
3.4 Networking

The network topology is flat. All clients can communicate directly with each MDS and DS node.

These nodes use standard TCP/IP, RDMA, or Infiniband networks, as desired. They also support such optimizations as GPUDirect Storage (GDS) to accelerate client performance.

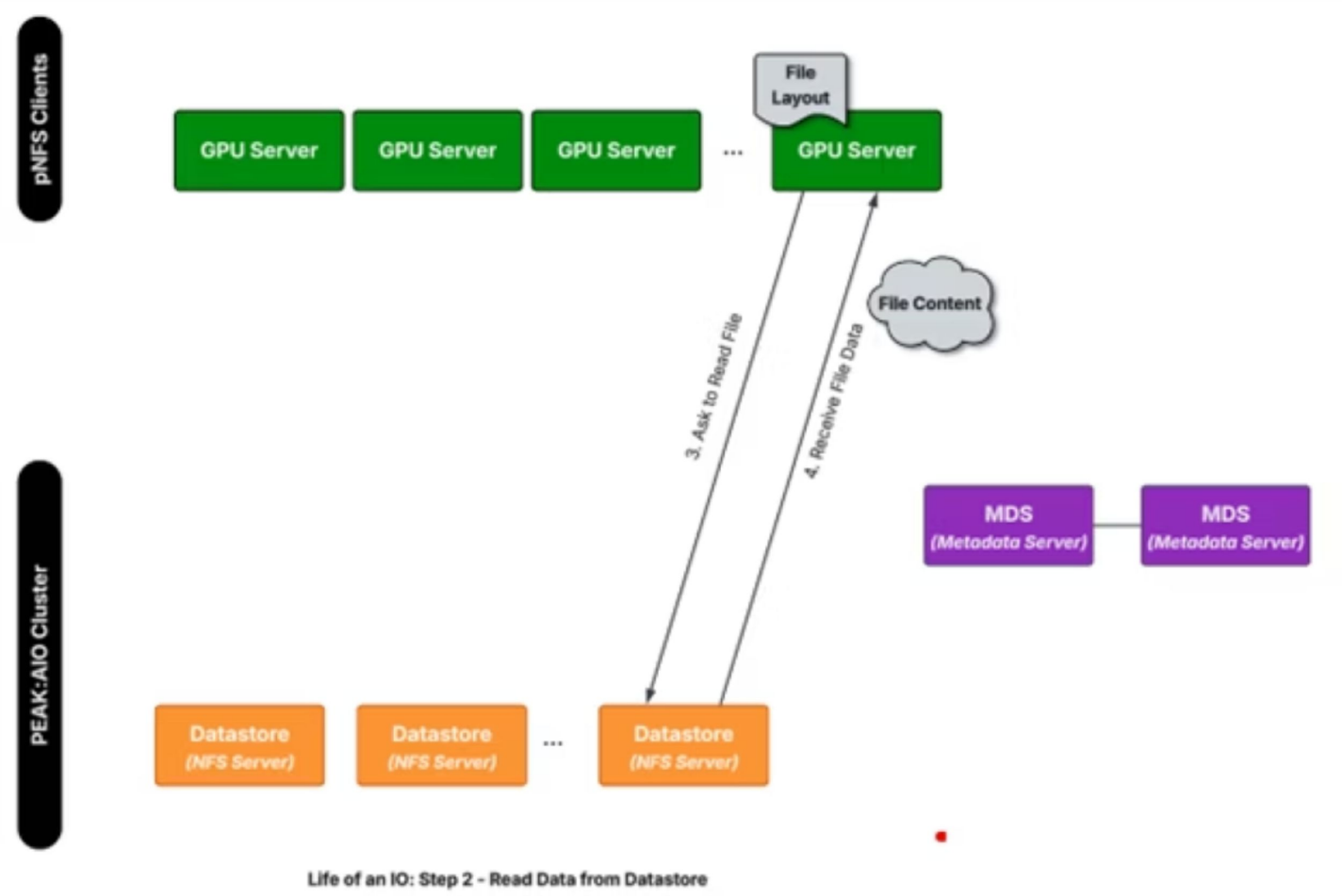
3.5 pNFS: Life of an IO

Now that the elements of the cluster have been covered, it is helpful to trace how the cluster handles IO and the role these elements play in optimizing this behavior. What is important to note is that while the MDS nodes aggregate information about where the files live, they do not participate in the major read/write IO path. When a client needs to read a file, it first contacts the MDS to get information about the file. The client is given a file layout, which is a pNFS structure that includes information about the file and where it is located. (By providing all of this information in one transaction, pNFS reduces the "chattiness" inherent in earlier versions of NFS.)



Life of an IO: Step 1 - Get Layout from MDS

Each client can communicate directly with each DS node in a mesh-like architecture. This means that clients have full bandwidth connections to all the DS nodes and don't have to proxy any of their read/write operations. Now that the client has the file layout, it can directly contact the appropriate DS node and perform the needed read/write operations at full network speeds.



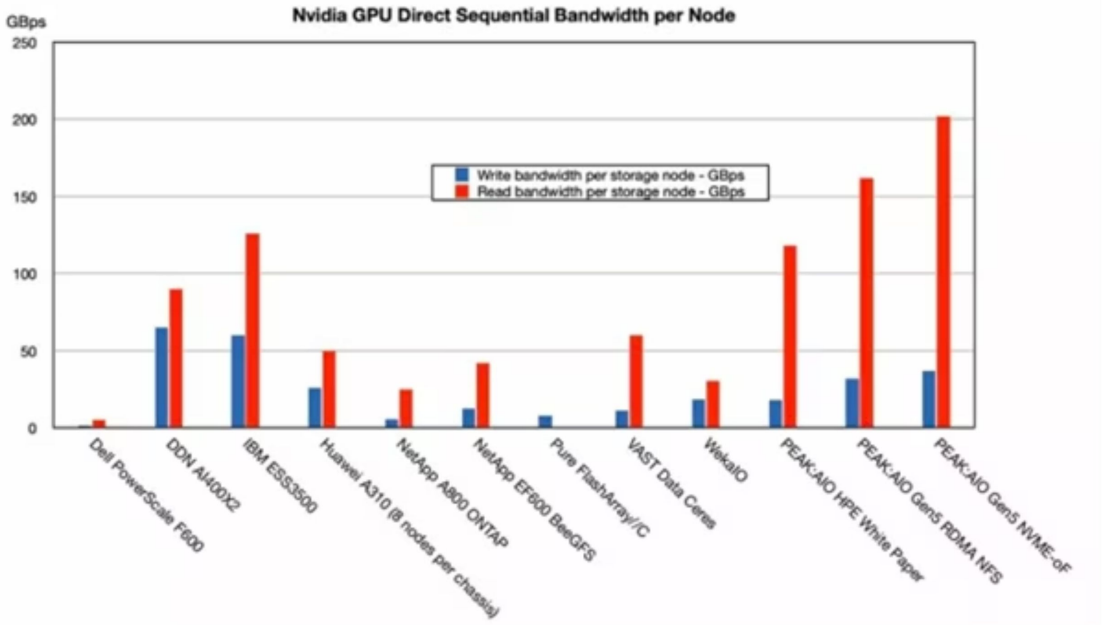
Life of an IO: Step 2 - Read Data from Datastore

This is different than traditional NFS scale-out architectures (e.g., Qumulo, VAST Data). In those architectures, clients will interact with one node in the cluster. This node will then conduct operations on the other nodes, often through a dedicated backend network, on behalf of the client, and then return the results. This “two hop” approach is inherently slower and less efficient, often limiting performance as the cluster scales (seen as a “knee in the curve” when scaling beyond a few nodes).

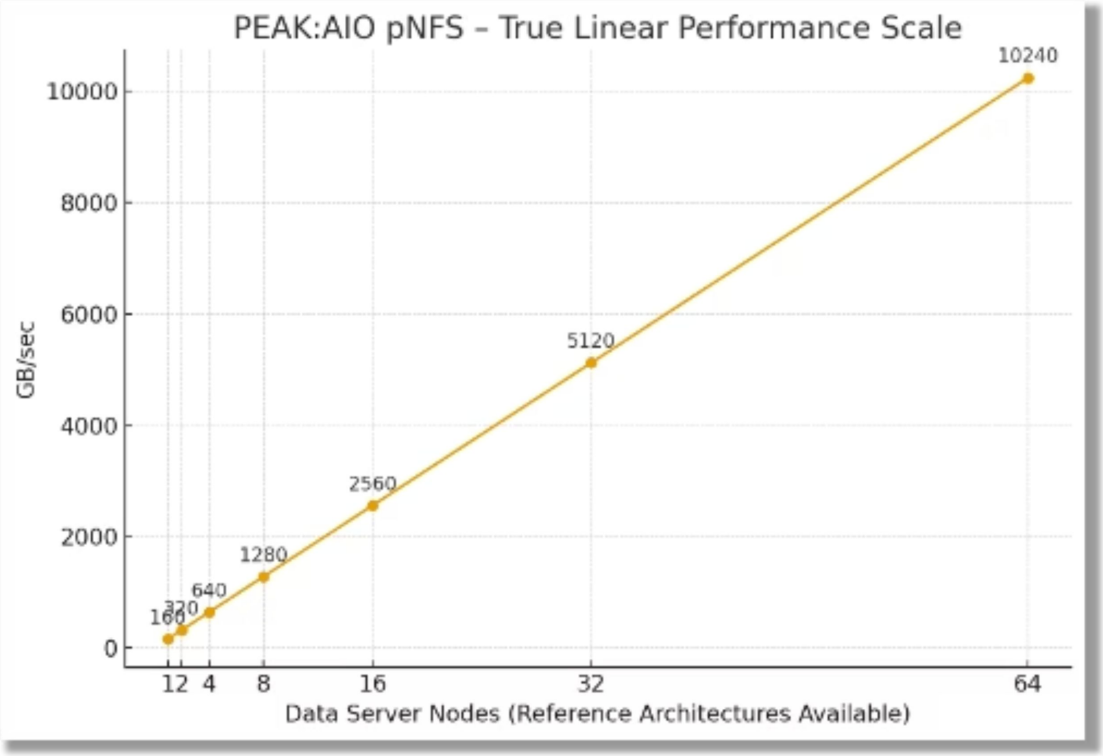
Section 4: Performance and Scaling

4.2 Datastore Performance and Scaling

A single PEAK:AIO datastore is optimized to get the most out of the underlying hardware as compared to many other solutions. This means it takes fewer nodes to get the performance required from a cluster. In this graph, we compare the number of nodes it takes to get comparable performance on a GPUDirect bandwidth benchmark. The per-node performance (especially on reads) does quite well vs the competition. This is achieved by optimizations in the NVMe, Linux, and NIC data paths.



As you need more performance (and/or capacity), more datastores can be added to expand the cluster. Once the new datastores can be added to the MDS, they will automatically become part of the namespace. This addition will be transparent to the user, and no modifications will be needed to the client. Because more drives and network connections have been added, there is more throughput available to each client. The graph below shows the scalability achieved by adding more datastores to the cluster.



4.2 MDS Scaling

Currently, MDS scaling is limited to an HA pair of nodes. Scaling metadata capacity is simply a matter of adding more drives. (This is trivial given the small size of metadata per file.)

While an MDS pair does a good job of scaling cluster performance as more DS nodes are added (as shown above), we are developing a sharded solution for scaling the MDS. Multiple MDS will be available to service requests, where each request will be directed to the appropriate MDS node based on the hash of the file path and name. This will allow you to add more MDS nodes to increase metadata performance.

4.3 Small File Workloads

Small file performance is a challenge with many workloads for clustered architectures. Two (batch) optimizations are currently under development to help accelerate these on PEAK:AIO clusters.

- File Pre-Creates
  - Empty files will be pre-created on each DS and registered with the MDS
  - When a new “file create” request arrives from the client, the MDS can assign it one of these pre-created files, making the request much faster
  - As the pre-create pool of files becomes smaller, a batch job to create a set of new files to repopulate this pool will begin in the background. This is a much more efficient means of creating these files.
- Write Batching
  - Writes of many small files can be written into a write buffer in high-speed storage.
  - These writes will be de-staged into the main storage asynchronously
  - This provides the client with a low-latency response while ensuring the writes are in stable storage.

Section 5: Failure Scenarios and High Availability

5.1 HA for Datastore Nodes

Data is protected within a DS using software RAID to protect against drive failures. To protect against a node failure, mirroring between nodes is currently required. This is accomplished by client-side mirroring (CSM) – a feature of pNFS where the client will issue 2 write requests to different DS nodes at the same time. The cluster can be configured to instruct clients to use CSM and will monitor client activity to ensure it is successful.

Development efforts are underway to provide a more space-efficient option: Client-Side EC. Using this option, clients will be instructed to write EC stripes across Datastore nodes (instead of mirroring). As with CSM, the MDS will monitor client activity and ensure operations are completed.

5.2 HA for MDS Nodes

MDS Nodes are deployed in HA pairs (active/passive). Writes to the metadata drive are written synchronously to the drives on the passive MDS as well. Node activity is monitored via an HA network connection. If the active MDS fails, the passive node will take over, using the data on its local (mirrored) drives.

Section 6: Managing Data Life Cycles

Putting data into a global namespace has many advantages when trying to manage storage policies. Data can be moved without impacting users. Here are a few ways this can be used to improve operations.

6.1 Tiering Data to Less Expensive Storage

It is convenient to store all of your data in one global namespace. Users don't have to know where the data lives to find it. However, where the data lives has cost implications if some of the storage is more expensive (and faster) than others. It is possible to have policies that manage what kind of storage various data lives on and still keep it under one global namespace. pNFS makes this possible.

Returning to the discussion about datastores, datastores can be pooled into different tiers by their storage type. High-performance NVMe datastores can be pooled into one tier, while slower NFS servers built with HDDs can be in a separate pool. These pools can be joined together under one global namespace by the MDS nodes, so users can see it as one namespace where the file path doesn't change when files get moved to different pools. (Including object storage as another tier is also under development.)

Policies can be used to down-tier unused data to slower tiers, while active data can stay on (or be promoted to) NVMe storage. All data movement is transparent to the user, even while they are accessing it.

6.2 Accelerating Writes on QLC

While reading QLC flash can be fast, there is a performance penalty when writing to it vs TLC. To help with this, a write buffer can be configured so that writes are written to higher-speed flash and then de-staged to QLC for longer-term use.

6.3 Simplifying Importing Data into the Cluster

When moving to a new storage solution, one of the biggest challenges is migrating the data. Data migration takes downtime and interrupts users' work. The PEAK:AIO cluster can help reduce the challenges with data migration off of existing NFS servers by dividing the process into two steps:

1. Using the feature called ***assimilation***, only the metadata needs to be replicated to the PEAK cluster. The data itself can remain on the original NFS server until a later date. Users can then mount the files on the global namespace and access their original files. Given that the amount of metadata is much smaller than the data, this can reduce the downtime from days to hours or less.
2. Once assimilated, files can be migrated transparently to datastores on the PEAK cluster, even while the files are being accessed. This allows you to complete the full migration over a longer period of time without impacting users.

Section 7: AI Workload Optimization

PEAK:AIO's performance architecture is not a one-size-fits-all design. Different AI and HPC workload profiles impose fundamentally different demands on storage systems, and a storage architecture optimized exclusively for one access pattern will deliver suboptimal performance for others. PEAK:AIO addresses this through workload-aware configuration and adaptive I/O management, tuning storage behavior dynamically to match the characteristics of the active workload.

Workload Type	I/O Characteristics	PEAK:AIO Optimization Strategy	Key Metric
Large Model Training	Sustained large sequential reads; multi-worker parallel access to shared datasets; periodic checkpoint writes	High-bandwidth streaming reads with aggressive read-ahead; parallel stripe allocation across maximum DS nodes; optimized checkpoint write coalescing	Aggregate sequential throughput (GB/s)
Checkpoint I/O	Burst parallel writes of large binary objects; high write concurrency; periodic overwrite of same logical paths	Parallel write striping; write-optimized NVMe allocation; layout pre-allocation to eliminate runtime MDS overhead during burst	Write throughput (GB/s); write latency (ms)
Inference Serving	Low-latency random reads of model weight files; high IOPS for batched request processing; read-mostly workload	Layout locality optimization to minimize DS hop count; NVMe read latency minimization; optional RDMA path for zero-copy model loading	Read latency (µs/ms); IOPS
Mixed AI + Analytics	Concurrent large streaming reads and small random I/O from analytics queries; variable file sizes; bursty access patterns	Intelligent workload classification; dynamic QoS to prioritize training I/O; parallel stripe allocation shared across workload classes	Mixed throughput; latency consistency under load
HPC Simulation	Highly parallel reads/writes from MPI processes; checkpoint/restart at scale; temporary scratch data volumes	MPI-IO optimized layout allocation; scratch tier fast-path; parallel writes across all DS nodes simultaneously	MPI-IO bandwidth; checkpoint restart time

These workload profiles are not static — a single cluster may transition between training, checkpointing, and inference phases within a single pipeline execution. The AI Storage Intelligence Engine continuously monitors I/O pattern signatures and adjusts optimization parameters in real time, ensuring that storage behavior tracks workload transitions without manual operator intervention.

Section 8: AI-Driven Storage Intelligence

The AI Storage Intelligence Engine is a first-class architectural component of PEAK:AIO Open pNFS — not an optional monitoring add-on. It represents PEAK:AIO's commitment to closing the loop between storage observability and operational action, transforming raw telemetry data into continuous, automated performance optimization and proactive infrastructure management.

Real-Time Telemetry Collection

The engine ingests high-frequency metrics from every layer of the PEAK:AIO stack: per-DS node throughput and IOPS, NVMe device-level latency histograms, MDS operation rates and queue depths, client-side I/O rates per GPU worker node, network fabric utilization per switch port, and NVMe device health indicators including write amplification factor, spare capacity, and uncorrectable error rates. This telemetry stream is processed in near-real-time to maintain a continuously updated performance model of the cluster.

Workload Profiling and I/O Pattern Recognition

Machine learning models trained on AI and HPC workload signatures analyze the telemetry stream to classify active workloads and predict their near-term I/O demands. Pattern recognition identifies the transition from sequential training reads to checkpoint write bursts, allowing the system to pre-position resources — such as pre-allocating layout stripes or adjusting read-ahead buffer sizes — before the workload transition occurs rather than reacting after performance has already degraded. This predictive posture is particularly valuable in large training runs where checkpoint events are scheduled at regular intervals and can be anticipated with high confidence.

GPU-to-Storage Correlation

The Intelligence Engine correlates storage I/O metrics directly with GPU utilization telemetry from NVIDIA DCGM or equivalent monitoring frameworks. This correlation enables operators to quantify precisely how storage latency spikes translate into GPU idle time — expressing storage performance problems in the currency of compute ROI rather than abstract I/O statistics. When GPU utilization drops below threshold, the system automatically investigates storage-side contributing factors and surfaces root-cause diagnostics.

Automated Hotspot Detection

Data access hot spots — conditions where a subset of DS nodes or NVMe devices receives disproportionate I/O load — are automatically detected and flagged for remediation. The engine can trigger data rebalancing operations to redistribute hot stripes across underutilized nodes, restoring balanced load distribution without requiring operator intervention or workload interruption. Hotspot detection also applies at the network layer, identifying switch port saturation conditions before they cause measurable throughput degradation.

Capacity Trend Forecasting

Time-series analysis of capacity consumption patterns enables the Intelligence Engine to project storage utilization trajectories and generate advance warnings when capacity thresholds will be reached. For AI training environments where dataset sizes grow with each model iteration and checkpoint accumulation can consume hundreds of terabytes per training run, capacity forecasting enables infrastructure teams to plan procurement and provisioning actions ahead of demand rather than responding to capacity alarms that interrupt active workloads.

Proactive Failure Detection

NVMe device health telemetry, combined with predictive failure models derived from device error logs, wear indicator trends, and latency anomaly patterns, enables the Intelligence Engine to identify devices likely to fail before they experience an unrecoverable error. Pre-failure detection triggers automated data migration off at-risk devices, replacement notification workflows, and, where applicable, proactive spare device promotion — ensuring that NVMe device failures do not result in data unavailability or performance degradation for active AI workloads.

Section 9: Deployment and Integration

9.1 Ethernet-Based Infrastructure

A significant operational barrier to deploying many high-performance parallel storage systems in enterprise data centers is their dependence on InfiniBand (IB) fabric infrastructure. InfiniBand delivers excellent latency and bandwidth characteristics for tightly coupled HPC workloads but introduces substantial operational complexity: dedicated IB switches and cabling plants, specialized HCA drivers and management tooling, operational staff with IB expertise, and fabric topologies that are physically and logically separate from the organization's standard Ethernet data center network. For enterprises deploying AI infrastructure within existing data centers — rather than building purpose-built HPC facilities — InfiniBand dependency represents both a capital cost barrier and an operational complexity multiplier.

PEAK:AIO Open pNFS is designed to deliver full parallel storage performance over standard high-speed Ethernet infrastructure. At 100GbE, a single DS node can sustain multi-GB/s throughput sufficient for most per-node AI workload demands. At 200GbE and 400GbE, DS nodes can saturate even the highest-throughput NVMe device configurations, ensuring that the network fabric never becomes the performance bottleneck. This Ethernet-native design means that PEAK:AIO clusters can be deployed on the same network infrastructure used for GPU cluster compute traffic, management, and storage — eliminating the dedicated IB plant and reducing both CapEx and OpEx.

100/200/400GbE

Full performance over standard Ethernet — no specialized IB switches, cabling, or HCAs required

Existing DC Fabric

Deploy on the same leaf-spine Ethernet fabric serving compute and management traffic — no parallel network build-out

Optional RDMA

RoCEv2 RDMA support available for CPU offload on latency-sensitive workloads — as an enhancement, not a prerequisite

9.2 Enterprise Integration and Deployment Flexibility

PEAK:AIO Open pNFS is designed to operate as a first-class storage tier within modern enterprise AI and HPC infrastructure stacks. Its NFSv4.2 standards compliance and Ethernet-native networking ensure broad compatibility with the orchestration, scheduling, and workload management platforms that govern enterprise AI operations.

Orchestration and Scheduler Integration

Kubernetes Environments: PEAK:AIO storage volumes are exposed as Persistent Volumes via the NFS CSI driver, enabling Kubernetes-native storage provisioning for containerized AI training and inference workloads. Storage classes can be configured to target specific performance tiers, with dynamic provisioning support for automated volume lifecycle management within MLOps pipelines.

Slurm Job Schedulers: For traditional HPC batch environments, PEAK:AIO integrates natively with Slurm through standard NFS mount points. Storage-aware job scheduling policies can be configured to ensure that jobs are dispatched only when sufficient storage bandwidth is available, preventing storage contention from degrading parallel job throughput.

Containerized AI Pipelines: PEAK:AIO storage paths are accessible from containers running on Docker, Podman, and containerd runtimes without modification. Container image layers, model artifact repositories, and training dataset volumes can all be hosted on PEAK:AIO, with shared access across multiple container instances enabled by the parallel multi-client architecture.

Deployment Configuration Models

On-Premises AI Clusters: Dedicated PEAK:AIO clusters co-located with GPU compute in enterprise data centers. Optimal for organizations with consistent, high-volume AI workloads requiring predictable performance and full data sovereignty.

Private Cloud Environments: PEAK:AIO integrates with OpenStack and VMware vSphere private cloud platforms, enabling storage-as-a-service provisioning models within enterprise cloud management frameworks. Storage QoS policies ensure multi-tenant workload isolation.

Research Environments: University HPC centers and research institutions benefit from PEAK:AIO's standards compliance and operational simplicity, enabling research teams to focus on scientific computing rather than storage administration.

AI Centers of Excellence: Centralized AI infrastructure hubs serving multiple business units benefit from PEAK:AIO's multi-workload optimization, enabling a single shared storage platform to simultaneously support training, inference, data engineering, and analytics workloads with appropriate QoS differentiation.

PEAK:AIO's hybrid cloud integration capabilities extend to support for tiered data management, where infrequently accessed datasets can be transparently migrated to object storage tiers (S3-compatible) while maintaining POSIX-compatible access semantics for AI framework compatibility. This tiering capability enables organizations to manage petabyte-scale dataset archives cost-effectively without sacrificing the ability to rapidly restore datasets to high-performance NVMe tiers when training runs require them.

Section 10: Reliability and Availability

Mission-critical AI infrastructure demands storage reliability that matches the value and time sensitivity of the workloads it serves. A failed storage system during an active multi-week model training run does not merely pause the workload — it can invalidate hours or days of compute investment if checkpoint data is unavailable for recovery. PEAK:AIO's reliability architecture is designed to ensure continuous availability of both the metadata path and the data path under component failure conditions, and to enable non-disruptive operational changes including capacity expansion and software maintenance.

PEAK:AIO's reliability architecture is built on five interdependent pillars that collectively ensure continuous availability across both the metadata and data planes. Redundant Metadata Services eliminate the MDS as a single point of failure through synchronous state replication and transparent failover. Distributed Data Redundancy protects against DS node loss through cluster-wide erasure coding and mirroring policies. Automatic Failover and Self-Healing mechanisms detect and recover from component failures without administrator intervention. Non-Disruptive Scaling and Maintenance procedures allow capacity expansion and software updates to be performed without pausing active workloads — a critical requirement in environments where multi-week training runs cannot tolerate planned downtime. And Snapshot and Data Protection Integration provides point-in-time recovery for training datasets and model checkpoints, with support for replication to secondary storage or cloud object storage for long-term retention and disaster recovery compliance.

1

Redundant Metadata Services

MDS instances are deployed in active/standby or active/active high availability configurations with synchronous state replication. Client lease state, layout mappings, and namespace modifications are durably committed to both MDS instances before acknowledgment, ensuring that a primary MDS failure results in transparent failover without client-visible I/O interruption. Layout leases held by clients remain valid across MDS failover events — active I/O to Data Servers continues uninterrupted.

2

Distributed Data Redundancy

Data stored across DS nodes is protected through erasure coding or mirroring policies configured at the file system level. Erasure coding schemes (such as 4+2 or 8+2 configurations) provide efficient space utilization with configurable fault tolerance — surviving simultaneous failures of multiple DS nodes without data loss. Unlike RAID implementations that protect only within a single node, PEAK:AIO's distributed redundancy spans the full cluster, ensuring that node failures are handled at the file system layer without requiring host-level RAID rebuilds.

3

Automatic Failover and Self-Healing

DS node failures are detected automatically through heartbeat monitoring. Upon detection, the MDS stops issuing layouts that reference the failed node, triggers data reconstruction from surviving redundancy group members, and begins distributing the recovered data across available healthy nodes. This self-healing process operates in the background without requiring administrator intervention and is designed to minimize the performance impact on active workloads by throttling reconstruction I/O during periods of high cluster load.

4

Non-Disruptive Scaling and Maintenance

DS nodes can be added to a running PEAK:AIO cluster without interrupting active workloads. The MDS begins incorporating new nodes into layout allocations immediately upon cluster join, and optional data rebalancing operations gradually migrate existing data to include new nodes in the distribution. Software updates to DS and MDS components are applied through a rolling update procedure that maintains full cluster availability throughout the maintenance window — critical for AI environments where training runs cannot be paused for planned maintenance events.

5

Snapshot and Data Protection Integration

PEAK:AIO supports file system-level snapshots for point-in-time recovery of training datasets and model checkpoints. Snapshots are crash-consistent and can be taken at the granularity of individual volumes or directories, enabling fine-grained recovery from accidental deletion or data corruption events. Integration with enterprise backup frameworks enables scheduled snapshot replication to secondary storage or cloud object storage for long-term retention and disaster recovery compliance requirements.

Section 11: Competitive Differentiation

PEAK:AIO Open pNFS occupies a distinct position in the enterprise AI storage market, addressing architectural limitations that persist across three categories of incumbent storage solutions: legacy NAS, proprietary parallel file systems, and general-purpose all-flash arrays. Understanding these differentiators is essential for infrastructure architects evaluating storage platforms against the demands of modern GPU-accelerated AI workloads.

vs. Legacy NAS

Legacy NAS platforms — including traditional enterprise filers and first-generation scale-out NAS— impose centralized controller architectures that are fundamentally incompatible with the parallel I/O demands of GPU clusters. Controller bottlenecks limit aggregate throughput regardless of the number of drives installed, and metadata serialization creates latency spikes that propagate across all connected clients simultaneously. PEAK:AIO eliminates these bottlenecks through pNFS-based direct parallel access, delivering performance that scales with data server count rather than controller capability. No software-defined NAS overlay or caching layer can restore the parallelism that a centralized control-plane architecture structurally prevents.

vs. Proprietary Parallel File Systems

Systems such as IBM Spectrum Scale (GPFS), Lustre, and BeeGFS offer genuine parallelism but impose significant operational overhead through proprietary client software stacks, specialized kernel modules, and complex administrative tooling. Client agent management across large GPU clusters — including version compatibility, kernel module rebuilding after OS updates, and cluster-wide upgrade coordination — represents a substantial and ongoing operational burden. PEAK:AIO delivers equivalent parallel I/O performance through the standard Linux NFSv4.2 kernel client, eliminating proprietary client lifecycle management while maintaining full standards compliance and operational simplicity

vs. General-Purpose All-Flash Arrays

General-purpose all-flash arrays (AFAs) — including platforms from Pure Storage, NetApp, and IBM — deliver strong single-node performance and enterprise feature sets well-suited to block and general-purpose file workloads. However, their architecture is optimized for vertical scaling and centralized controller performance, not for the horizontal, multi-client parallelism that AI/HPC demands. As GPU cluster size grows, AFA controller throughput becomes a fixed ceiling: adding more GPU nodes increases I/O demand but cannot increase the storage throughput available to serve it. The result is the same GPU starvation dynamic that characterizes legacy NAS, simply deferred to a higher node count before the bottleneck becomes visible.

PEAK:AIO's pNFS removes this ceiling entirely. Where an AFA routes all I/O through a fixed controller tier, PEAK:AIO distributes I/O directly across an expandable pool of NVMe Data Server nodes — each contributing independent network bandwidth and NVMe throughput to the aggregate. The performance available to each GPU client scales linearly with DS node count, with no controller upgrade cycle required to accommodate cluster growth. For organizations that have invested in AFA infrastructure for general workloads, PEAK:AIO complements rather than replaces that investment: AFAs continue to serve block and latency-sensitive workloads effectively, while PEAK:AIO delivers the parallel throughput that AI training and HPC workloads specifically require.

Section 12: Use Cases

PEAK:AIO Open pNFS addresses a broad spectrum of GPU-accelerated and HPC workload profiles. The following use cases represent the primary deployment scenarios where the parallel architecture delivers the most significant performance and operational advantages over incumbent storage solutions.

1

AI Model Training

Distributed deep learning training workloads are the primary design target for PEAK:AIO. Training runs for large language models (LLMs), vision transformers, and multimodal models distribute computation across dozens to hundreds of GPU nodes, each simultaneously reading mini-batch data from shared storage at high bandwidth. PEAK:AIO's parallel architecture ensures that storage bandwidth scales with GPU count — every GPU worker accesses data via a direct, parallel path to DS nodes, with no shared bottleneck that would cause GPU starvation as cluster size grows. Checkpoint I/O — periodic writes of model weights and optimizer state that can reach hundreds of gigabytes per event — is handled through parallel write paths that sustain high write throughput without interrupting ongoing read operations on other nodes.

2

HPC Research Workloads

Scientific simulation workloads — including computational fluid dynamics, molecular dynamics, climate modeling, genomics pipelines, and finite element analysis — generate I/O patterns that closely parallel AI training demands: highly parallel reads and writes from MPI processes, large checkpoint/restart datasets, and temporary scratch volumes that require extremely high write throughput during simulation execution. PEAK:AIO's MPI-IO-compatible parallel access model and high-bandwidth write paths make it well-suited for HPC research environments, while its Ethernet-native operation simplifies deployment in university and national laboratory data centers that may not maintain InfiniBand infrastructure.

3

Enterprise Analytics Pipelines

Data engineering and analytics workloads that feed AI model training pipelines — ETL processing, feature engineering, dataset preprocessing, and model evaluation — require storage that can sustain concurrent read and write access from multiple pipeline stages simultaneously. PEAK:AIO's multi-workload optimization capabilities allow data engineering pipelines and active training runs to share the same storage cluster without mutual performance interference, enabled by intelligent I/O classification and QoS management within the AI Storage Intelligence Engine.

4

Media and Content Rendering

Visual effects and computational media production environments operate parallel rendering farms where hundreds of render worker processes simultaneously read scene data, texture libraries, and asset files from shared storage. These workloads demand the same combination of high aggregate bandwidth and parallel client access that characterizes AI training, making PEAK:AIO an effective storage platform for production rendering pipelines alongside AI and HPC workloads. Mixed-use deployments that combine AI model training with rendering workloads benefit from PEAK:AIO's workload classification and QoS capabilities to ensure rendering deadlines are met without compromising training throughput.

Section 13: Operational Benefits

The operational impact of PEAK:AIO Open pNFS extends beyond raw storage performance. Infrastructure architects and operations teams responsible for AI platform management face a dual mandate: maximize GPU utilization and model training throughput while minimizing operational complexity, unplanned downtime, and total cost of infrastructure ownership. PEAK:AIO's architectural design and AI-driven intelligence layer address both dimensions simultaneously.

13.1 Improved GPU Utilization

By eliminating storage-induced GPU starvation through parallel, high-bandwidth data paths, PEAK:AIO directly increases the fraction of provisioned GPU compute time spent on model computation versus waiting on I/O. In large-scale training environments where GPU-hours are the primary cost driver, even modest improvements in GPU utilization translate into substantial reductions in time-to-convergence and effective cost-per-model. The AI Storage Intelligence Engine's GPU-to-storage correlation capability enables operations teams to quantify this utilization improvement and demonstrate its financial impact to infrastructure investment decision-makers.

13.2 Faster Time-to-Model Convergence

Reduced storage latency and higher sustainable I/O throughput enable AI training frameworks to maintain higher data ingestion rates, reducing the wall-clock time required to complete a full training epoch. Across a training run spanning dozens of epochs and multiple days of compute time, consistent storage performance improvements compound into meaningful reductions in total training duration — enabling faster model iteration cycles, shorter time-to-production for AI-driven features and services, and more efficient utilization of GPU cluster reservation windows.

13.3 Reduced Infrastructure Sprawl

PEAK:AIO's multi-workload optimization and QoS management capabilities allow a single converged storage cluster to serve training, inference, analytics, and general data management workloads concurrently. This consolidation eliminates the need to provision separate, purpose-specific storage systems for each workload category — reducing the number of discrete storage platforms to procure, deploy, monitor, and maintain.

Consolidated storage also simplifies data management by eliminating the need to stage datasets between separate storage systems as they move through the AI pipeline from raw ingestion through training to model serving.

13.4 Simplified Management

Standards-based NFS access eliminates the proprietary client management overhead associated with parallel file systems. Linux kernel NFSv4.2 client support is included in all major enterprise Linux distributions and requires no additional installation, configuration, or lifecycle management. The AI Storage Intelligence Engine provides a single management interface for monitoring cluster performance, capacity, and health — with automated remediation of common operational issues reducing the frequency of administrator intervention. Non-disruptive scaling and rolling maintenance procedures minimize planned maintenance windows and eliminate unplanned downtime.

13.5 Predictable Scaling Economics

PEAK:AIO's linear performance scaling model enables straightforward capacity and performance planning. Operators can project the throughput impact of adding DS nodes with high confidence, aligning procurement decisions with AI project scaling roadmaps rather than discovering performance plateaus only after infrastructure investment has been made. This predictability reduces the risk of over-provisioning storage at initial deployment to accommodate anticipated future scale, enabling a more gradual, demand-driven expansion strategy that aligns capital expenditure with demonstrated business demand.

Section 14: Business Impact

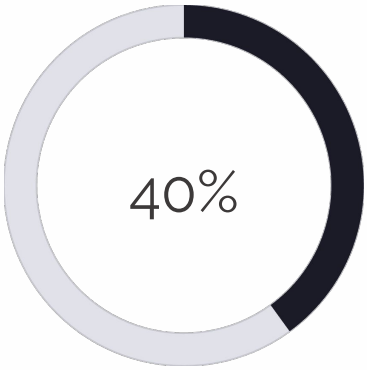
The infrastructure benefits of PEAK:AIO Open pNFS translate directly into measurable business outcomes for organizations operating AI platforms at scale. The relationship between storage architecture and business impact is most clearly expressed through three financial lenses: infrastructure ROI, time-to-value for AI initiatives, and total cost of ownership across the full AI platform stack.

Accelerated AI Project Timelines

Storage-induced training delays have a direct and quantifiable impact on AI project schedules. When GPU clusters idle waiting on data, training runs that could complete in two weeks instead require three — a 50% schedule extension that delays model deployment, postpones business value realization, and consumes GPU reservation budget without proportional output. PEAK:AIO's parallel architecture eliminates this delay multiplier, enabling training schedules to reflect actual GPU compute time rather than compute time plus storage wait time. For organizations managing large portfolios of AI initiatives, consistent training schedule reliability enables more accurate project planning and more efficient GPU cluster utilization across concurrent projects.

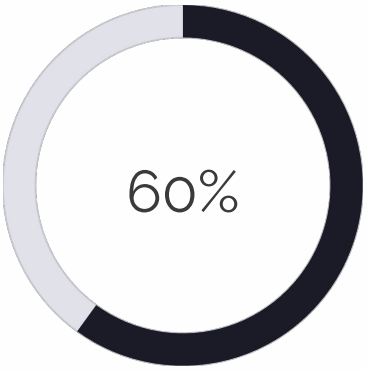
Higher Infrastructure ROI

GPU clusters represent the dominant capital expenditure line item in enterprise AI infrastructure budgets. NVIDIA H100 and next-generation GPU nodes are among the most expensive compute infrastructure investments an organization makes, with individual DGX systems priced in the hundreds of thousands of dollars. When storage architecture limits GPU utilization to 60-70% of provisioned capacity, the effective cost-per-useful-GPU-hour increases by 40-60% relative to a fully utilized cluster. PEAK:AIO directly improves this utilization metric, recovering infrastructure ROI that would otherwise be sacrificed to storage bottlenecks. In large-scale AI infrastructure deployments, the ROI improvement attributable to higher GPU utilization alone can exceed the total cost of the storage platform within the first year of operation.



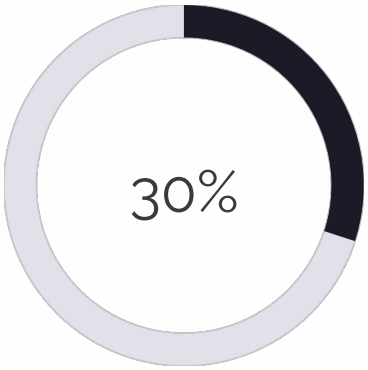
GPU Utilization Loss

GPU utilization recovered by eliminating storage-induced I/O bottlenecks in distributed AI training environments — directly reducing effective cost-per-GPU-hour



Throughput Increase

Aggregate storage throughput increases versus centralized NAS at equivalent node count, enabling more GPU workers to be served simultaneously without I/O contention



OpEx Reduction

Reduction in storage operational overhead from eliminating proprietary client lifecycle management, consolidating workload tiers, and automating failure remediation

Organizations that align their storage architecture with GPU-scale computing do not merely solve a technical problem — they unlock the full business value of their AI infrastructure investments. PEAK:AIO Open pNFS provides the storage foundation that enables AI programs to scale from proof-of-concept to enterprise production without storage architecture becoming the factor that limits ambition or extends timelines.

Section 15: Conclusion

AI workloads demand a fundamentally different storage architecture than the enterprise data management workflows that NAS systems were designed to serve. The performance characteristics of modern GPU clusters — extreme I/O concurrency, high aggregate bandwidth demands, mixed large-file streaming and small-file random access, and tolerance for nothing less than sustained throughput — are structurally incompatible with centralized NAS architectures, regardless of how aggressively those architectures are tuned or how modern their underlying hardware.

PEAK:AIO Open pNFS resolves this incompatibility at the architectural level, not through software workarounds or caching overlays, but through a ground-up parallel architecture built on the NFSv4.2 pNFS standard. By separating the metadata path from the data path, distributing data I/O directly across NVMe-optimized Data Server nodes, and enabling every client to operate in parallel against every storage node simultaneously, PEAK:AIO delivers the throughput, scalability, and latency characteristics that GPU-accelerated AI workloads require.

True Parallel Data Access

Direct client-to-DS I/O via pNFS layout distribution — no centralized data controller in the data path

Linear Scalability

Aggregate throughput and capacity scale proportionally with each additional Data Server node — no architectural ceiling

NVMe-Optimized Performance

PCIe Gen4/Gen5 NVMe storage tiers deliver sub-millisecond latency and multi-GB/s per-node sustained throughput

AI-Driven Intelligence

Continuous telemetry, workload profiling, and predictive optimization aligned to GPU workload demands

Enterprise-Grade Reliability

Redundant MDS, distributed erasure coding, automatic failover, and non-disruptive scaling for mission-critical AI environments

By leveraging open standards, advanced architectural design, and integrated AI-driven storage intelligence, PEAK:AIO provides a future-ready storage foundation capable of supporting the full lifecycle of enterprise AI, ML, and HPC initiatives — from initial research exploration through production-scale model deployment. Infrastructure architects and storage engineers evaluating GPU-accelerated AI storage solutions will find in PEAK:AIO Open pNFS a platform that eliminates the fundamental storage constraints that have historically limited AI infrastructure ROI, and that scales to meet the demands of AI workloads as they continue to grow in complexity, scale, and strategic importance.

For technical specifications, proof-of-concept engagement, and detailed performance benchmarking data, contact your PEAK:AIO solutions architect or visit the PEAK:AIO technical documentation portal.