Tech Brief: NeuralMesh Fast S3 Object Storage

Architected for the performance and scale of AI infrastructure

Table of Contents

Abstract

S3 is the default protocol for AI infrastructure. Conventional S3 object store implementations were built for sequential, moderate-concurrency access, not the checkpoint-intensive training runs, concurrent inference requests, and millions of operations that GPU clusters generate continuously. The result is a performance ceiling that shows up directly in GPU utilization rates and a structural data copy tax imposed by running file and object storage as separate systems.

NeuralMesh delivers the distributed architecture AI infrastructure demands: no centralized controllers, native S3 and POSIX on the same physical data blocks, and the S3 Performance Bucket, purpose-built optimizations for the operations AI pipelines run most. That architecture holds under GPU cluster concurrency instead of degrading: more than 2x the nearest published ListObject throughput (7.5M ops/sec) and 5x the conventional limit on concurrent connections per node (5,000 versus roughly 1,000) keep request queuing from cascading through every job on the cluster. On the next-generation WEKApod systems announced today, that same architecture reaches 3 to 4x greater rack-scale throughput than the nearest published alternative (10.2 TB/s and 210 million IOPS per rack, WEKApod Nitro), rack-scale numbers no general-purpose hardware platform can match. Performance scales linearly as nodes are added with no architectural ceiling.

S3 is the default protocol for AI infrastructure. Every major framework supports it. Every pipeline assumes it. Most S3 object store implementations were designed for a different world, one where object storage sat at the edge of the pipeline, durable, scalable, and accessed sequentially by a modest number of clients. That world is gone. Training jobs now checkpoint to S3 hundreds of times per run. Inference services pull model weights at request time under concurrency from thousands of simultaneous sessions. Agentic workflows generate millions of small objects per session. Object storage is now in the middle of every AI pipeline stage, handling traffic patterns it was never designed for.

NeuralMesh supports 5x the conventional limit on concurrent connections per node (5,000 versus roughly 1,000) and distributes metadata across every node in the cluster with no shared connection pool and no centralized metadata controllers. Under AI workload concurrency, hundreds of simultaneous training and inference jobs generating continuous ListObject calls, PUT requests, and catalog scans, performance holds. Request queuing doesn't cascade. PUT latency stays flat. Every GPU cycle bills or produces output. Storage stops being the constraint.

Redundant copies are an architecture problem

A typical AI pipeline carries multiple full dataset copies before a model ships, sometimes two or three, sometimes five or more, depending on the pipeline. Each copy is a sequencing dependency. Downstream stages cannot begin until it completes. Each one temporarily doubles storage footprint. At petabyte scale, copies that take hours become the critical path in pipelines measured in days. The copy step is not a workflow inefficiency. It is structurally required when file and object run on separate systems with no shared data layer.

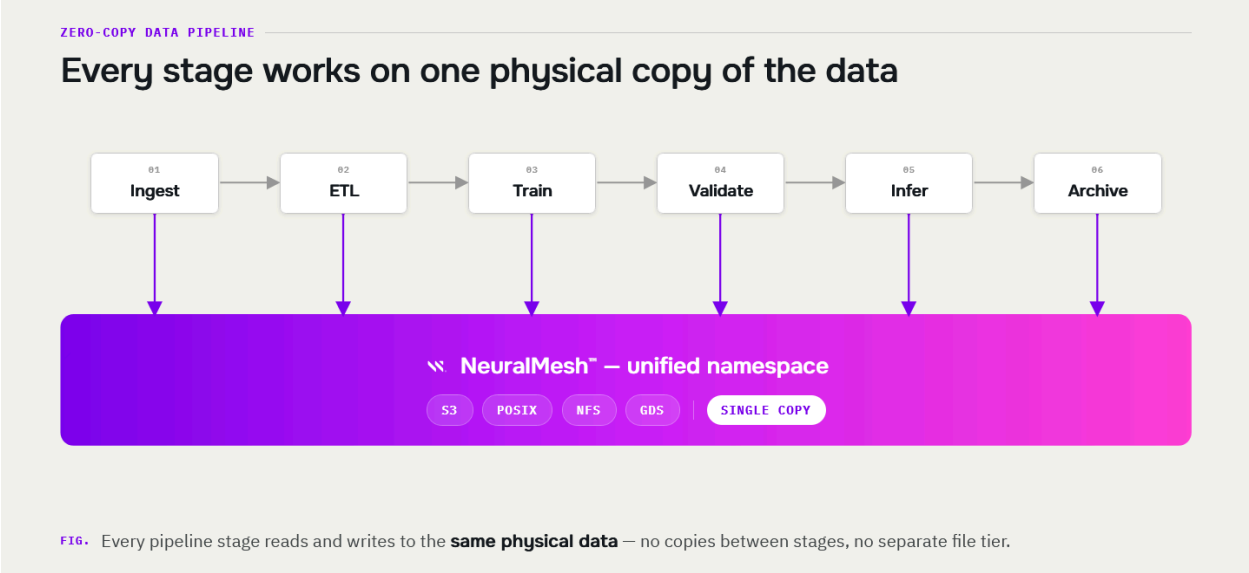

FIG. Every pipeline stage reads and writes to the **same physical data** — no copies between stages, no separate file tier.

NeuralMesh Object Store: Built without ceilings

NeuralMesh addresses both failure modes through three foundational architectural decisions. These are not optimizations applied to a conventional storage stack. They are the properties the system was designed around from the start.

Fully distributed data and metadata.

In conventional architectures, metadata ownership sits in a small number of dedicated servers. As clusters grow and concurrency increases, those servers become the constraint. NeuralMesh distributes both data and metadata across every node using deterministic hash-based placement. Any node can locate and service any request without coordination. There are no dedicated metadata servers to become bottlenecks and no centralized coordination that serializes parallel workloads. Metadata performance scales linearly with the cluster.

S3 Performance Bucket.

Standard S3 implementations carry protocol overhead that compounds under AI workload patterns, specifically enumeration, high-concurrency writes, and model weight retrieval under inference load. The S3 Performance Bucket is a set of targeted protocol-level optimizations addressing exactly those operations. The detail of each optimization is in the following section.

Native S3 and POSIX on the same physical data blocks.

NeuralMesh is not an S3 gateway layered over a file system, nor a file system with an S3 adapter. Both protocols are first-class implementations operating directly on the same underlying data blocks. A file written via NFS is immediately readable via S3. An object written via S3 is immediately accessible via POSIX. There is no translation layer, no replication lag, and no consistency gap between access modes.

Parallel data striping via RDMA/SR-IOV

When a client reads or writes data, NeuralMesh distributes the operation in parallel across every node in the cluster simultaneously. No designated data node targets. No single node that can become a bottleneck. Clients communicate with the entire cluster concurrently over RDMA, bypassing CPU overhead on the data path entirely. Throughput scales with nodes, not with the capacity of any individual component.

S3 optimized for the operations AI pipelines actually run

On top of that architectural foundation, the S3 Performance Bucket addresses the protocol-level overhead that standard S3 implementations carry by default — targeting the exact operations AI pipelines run continuously: catalog enumeration, high-concurrency writes, and model weight retrieval under inference load.

Fast ListObject enumeration

Replaces sequential catalog scanning with parallel enumeration distributed across every node simultaneously. Result: 7.5 million ListObject operations per second and 281 million files enumerated in 13.6 seconds — the operation pattern that causes conventional metadata controllers to saturate first under AI load.

Data Integrity Freedom

Decouples content verification from the S3 API surface for workloads that manage their own data integrity externally. AI training frameworks typically implement checksum validation within the pipeline. Removing the redundant verification layer reduces per-operation overhead without sacrificing data integrity.

Lightweight ETAG hashing

Replaces default MD5-based ETAG computation with a faster hashing scheme that reduces CPU overhead on PUT operations at high concurrency. In workloads generating continuous checkpoint writes from hundreds of simultaneous workers, the savings translate directly to lower PUT latency and higher sustainable write throughput.

S3 over RDMA

Replaces TCP as the transport layer in RDMA-capable environments, enabling zero-copy data transfer directly to GPU memory. For inference workloads loading model weights from S3 at request time, this eliminates CPU and memory overhead from the conventional TCP data path. Model weight load latency drops. Time-to-first-token improves.

What the architecture delivers

The architectural decisions above produce measurable outcomes at AI infrastructure scale. These results come from production deployments, not controlled benchmark conditions.

Outcome	Result	What it means
Concurrent connections per node	Up to 5,000 vs. ~1,000 for conventional S3 object stores	GPU clusters scale past the connection ceiling that causes cascading queuing on conventional platforms
S3 Performance Bucket improvement	500–600% over standard S3 configuration	Protocol-level overhead eliminated on the operations AI pipelines run most — ListObject, PUT, and weight retrieval
Model training time	2x faster across 30M concurrent project files	Storage layer removed from the training critical path in a production AI drug discovery environment
GPU output	4.2x output per GPU in production deployment	GPU utilization recovered directly from elimination of storage-side queuing and latency

The architecture that gets stronger as it grows

The performance figures above describe a deployment at a point in time. The more important question for infrastructure architects is whether the architecture that delivers those results at six nodes delivers them the same way at six hundred.

Most object storage architectures have a centralized component in the metadata path. These components perform well at moderate scale and become the bottleneck as cluster size and concurrency increase together. Operators add nodes and observe that throughput does not scale proportionally — because the metadata path, not the data path, is the constraint. NeuralMesh has no centralized metadata component to become that bottleneck. Deterministic hash-based placement means every node can locate every piece of data without coordination. Performance scales linearly with nodes. There is no architectural ceiling that flattens the curve.

Resilience follows the same logic and compounds at scale rather than degrading. Every node is an independent failure domain. WEKApod's cable-based, backplane-free drive interconnect creates independent failure domains at the drive level, a property that requires co-designing the chassis and software together and cannot be replicated on general-purpose hardware. A single drive failure stays a single drive failure: no propagation, no cluster-level intervention, no impact on running workloads. As the cluster grows, the number of independent failure domains grows with it. The system becomes structurally harder to disrupt, not more fragile.

NeuralMesh is in production at sovereign AI providers managing multi-petabyte deployments with thousands of concurrent GPU clients across Malaysia, India, and Thailand. The architecture that delivered consistent performance at initial deployment continues to deliver it as those deployments have grown.

One data store. Every protocol. No copies.

Eliminating the performance ceilings addresses the first failure mode. Eliminating the copy tax requires a different architectural decision: running S3 and POSIX natively on the same physical data blocks.

NeuralMesh Fast S3 Object Storage is not an S3 gateway bolted onto a file system. S3 and POSIX are both first-class protocol implementations operating on the same underlying data blocks. A file written through NFS is immediately readable through S3. An object written through the S3 API is immediately accessible through POSIX. There is no translation layer, no replication lag, and no consistency gap between access modes. A data ingestion pipeline writing via S3, a GPU training container reading via POSIX, and a RAG pipeline serving embeddings back via S3 all operate against the same data, at every stage, without copies between them.

Cross-protocol governance applies the same policy layer across every access mode from a single control plane. S3 lifecycle rules, IAM policies, encryption, and audit logging defined once apply consistently to data accessed through S3, POSIX, NFS, and SMB. A lifecycle rule set in S3 applies to data written via POSIX. For organizations with compliance requirements spanning multiple data access patterns, this eliminates the per-protocol policy gaps that create governance drift in conventional multi-tier architectures.

Data management for production AI pipelines

NeuralMesh includes the data management layer required to run a complete production AI pipeline, not as add-on capabilities but as first-class features of the platform.

Granular lifecycle rules

S3-compatible object lifecycle management with automatic expiration or transition based on prefixes, object tags, or age. Rules apply consistently regardless of the access protocol used to create the object. A file written via POSIX can be expired by an S3 lifecycle policy. Eliminates the governance gaps common in hybrid storage environments.

Event-driven pipeline integration

Push-based event notification to user-defined Kafka targets on object create, modify, or delete. Downstream systems, including AI preprocessing workers, vector embedding generators, and data indexers, react to data changes in real time without polling latency. Fully event-driven pipeline architecture on standard enterprise streaming infrastructure.

Object versioning

Every version of every object is preserved, retrievable, and restorable. Overwriting creates a new version; deleting creates a delete marker. Deterministic recovery from human error or application faults. This is a hard requirement in AI pipelines where training datasets and model checkpoints must be reproducible.

Object tagging and pre-signed URLs

Key-value metadata tags on any object drive lifecycle policies, cost allocation, and data organization. Pre-signed URLs support temporary, credential-free access for secure data sharing with external compute environments, auditors, or partner systems without exposing permanent credentials.

Security enforced across every protocol, from one control plane

Security is enforced consistently across every access protocol — S3, POSIX, NFS, and SMB — from a single control plane. No protocol-specific gaps, no separate credential silos, no permission drift.

IAM policies

Fine-grained S3 operation and resource permissions at the user level, consistent with standard IAM semantics. Predefined or custom JSON policies apply at bucket and prefix granularity.

Temporary credentials via STS

Zero-trust access patterns for ephemeral compute environments. AssumeRole with optional reduced-capability policies enables secure temporary access without long-lived credentials.

Forensic audit logging

Every S3 API call streamed as a structured JSON event via webhook to SIEM platforms. Each event captures requester identity, network metadata, object details, and operation status — complete visibility into every data access for compliance reporting and security investigation.

Encryption

TLS 1.2+ for all data in transit. Filesystem-level encryption for all data at rest, enforced regardless of access protocol — S3, POSIX, NFS, or SMB.

LDAP-based S3 identity

Existing LDAP credentials map to dynamically generated S3 access keys. IAM policies map directly from LDAP user attributes. Consistent identity across every protocol with no separate credential silo.

Consistent UID/GID identity mapping

UID and GID permissions set at the POSIX layer are respected by S3 and vice versa. Eliminates permission drift in multi-protocol environments where the same data is accessed through different protocol surfaces.

Conclusion

NeuralMesh Fast S3 Object Store is a set of decisions made at the architecture level, not add-on optimizations. Distributed metadata, no centralized controllers, and native S3 and POSIX on the same physical data blocks are why performance holds under GPU cluster concurrency, and why file and object storage stop requiring separate systems. The same architecture runs the same way at six nodes and at six hundred, on-premises, in hyperscale, hyrbid, or AI Clouds, on commodity hardware or on WEKApod.

See NeuralMesh Fast S3 Object Storage in action. Request a technical deep-dive or benchmark walkthrough at weka.io/contact.

WEKApod™ + NeuralMesh Fast S3 Object Storage:

The most dense AI object storage platform available

Purpose-built hardware and software, designed together from the start:

- **Over 1 exabyte effective capacity per rack** — more usable storage per rack unit than any platform built on general-purpose hardware
- **10.2 TB/s throughput** — full performance sustained under real AI workload concurrency
- **QLC economics without QLC penalties** — AlloyFlash steers writes by I/O type so the highest-capacity configurations are production-viable, not just on a spec sheet
- **Up to 6x data reduction on AI/ML datasets** — always-on deduplication and compression on AI training data, with write overhead under 5%
- **NVIDIA NCP-certified** — validated configurations, rack to production without integration work