FSx for Lustre vs. WEKA NeuralMesh: Four Breaking Points

A decision-ready comparison for AI infrastructure architects evaluating AWS storage options

Challenges

- Throughput ceiling hits before capacity fills
- Single-AZ default exposes production workloads
- No persistent KV cache for inference
- Multi-tenancy without QoS is unenforceable

Solution

NeuralMesh delivers decoupled throughput, multi-AZ resilience, persistent KV cache, and native QoS — purpose-built for GPU-era AI workloads.

Benefits

- Throughput scales independently of capacity
- Replication and backup across AZs and regions
- Performant single-namespace tiering to S3
- Persistent key-value cache for inference
- Per-tenant QoS enforcement at scale

FSx for Lustre was built for traditional HPC scratch space. It was a reasonable choice when AI training jobs were short-running and single-tenant. Modern AI infrastructure is neither of those things.

Today's GPU clusters need sustained throughput across long training runs, persistent key-value cache for inference serving, multi-AZ resilience for production uptime, and the ability to enforce per-tenant SLAs when multiple model teams share infrastructure. FSx for Lustre hits walls on all four.

This guide compares FSx for Lustre and WEKA® NeuralMesh™ across four dimensions where AI teams consistently feel constrained – with Stability AI's production deployment on AWS as the reference proof point.

BREAKING POINT #1

Throughput-capacity coupling

FSx for Lustre's Persistent_2 tier delivers 1,000 MB/s per TiB of provisioned capacity. To reach 17 GB/s sustained throughput, you must provision 19.2 TiB — regardless of how much storage your workload actually needs. If your dataset is 6 TiB, you're still paying for 19.2 TiB to keep the GPUs fed.

NeuralMesh decouples throughput from capacity. You configure your throughput target independently of your storage volume. The two scale independently. Stability AI used this to reach 15x autoscale headroom on a 400-node AWS cluster without re-provisioning storage capacity.

	FSx for Lustre	WEKA NeuralMesh
Throughput model	Choose one of four fixed performance levels per TiB – 125, 250, 500, or 1,000 MB/s (coupled)	Decoupled — configure independently
Capacity constraint	May have to over-provision to hit throughput targets	Provision the capacity your data requires
Autoscale behavior	Manual re-provisioning cycle required	Elastic — scales without re-provisioning

---

BREAKING POINT #2

Resilience and availability

FSx for Lustre is single-AZ by default. Production AI training and inference workloads that require cross-AZ resilience need additional architecture — and additional cost — to achieve it. A single-AZ failure during a multi-day training run means starting over.

NeuralMesh can be deployed natively across any AZ or region — but for most production workloads, built-in replication and backup deliver resilience without the cost of running active multi-AZ infrastructure. NeuralMesh also supports performant single-namespace tiering to S3, plus on-premises to cloud data movement through a single consistent data layer — which is how Scenario B (cloud repatriation) teams move workloads without rebuilding the storage architecture.

	FSx for Lustre	WEKA NeuralMesh
AZ resilience	Single-AZ default; multi-AZ requires extra config	Deployable natively across any AZ; replication and backup built in
Multi-region	Not supported natively	Deployable across regions on the same data layer, with performant tiering to S3
On-prem portability	Not applicable	Consistent data layer across cloud and on-prem

BREAKING POINT #3

Inference readiness

FSx for Lustre has no persistent key-value (KV) cache layer. Inference workloads that need to serve cached attention states across concurrent requests must build and manage a separate caching tier — typically Redis or a custom in-memory layer — sitting in front of storage.

That architecture works at low concurrency. Under load, it breaks in two ways: every cache miss travels through an additional network hop back to the storage layer, adding latency at exactly the moment when requests are competing for attention. And the separate cache system becomes an independent operational surface — with its own invalidation logic, failure modes, and capacity planning.

NeuralMesh includes a persistent KV cache at the storage layer itself. Attention states are cached natively — no separate tier, no additional network hop, no second system to operate. Firmus measured 4x faster time to first token (TTFT) in production on a NeuralMesh-backed inference cluster under concurrent load that would have saturated a separate caching layer.

NeuralMesh also accelerates model checkpoint and model loading for inference. Because WEKA performance lets GPUs read directly against storage — instead of offloading checkpoints or staging models to local NVMe first — teams get model checkpoint acceleration, transaction latency in the hundreds of microseconds, and a central model repository their GPUs can pull from directly.

	FSx for Lustre	WEKA NeuralMesh
KV cache	None — requires a separate caching layer	Persistent KV cache native
TTFT under load	Degrades as concurrent requests increase	Sustained via cached attention states
Inference architecture	Additional tier required; increases latency	Single-layer — storage handles caching

	FSx for Lustre	WEKA NeuralMesh
KV cache	None — requires a separate caching layer	Persistent KV cache native
Model checkpoint & loading	Requires offloading checkpoints or staging models to local NVMe	Direct GPU access to storage, no staging, hundreds-of-microsecond latency

BREAKING POINT #4

Multi-tenancy and QoS

Platform teams running multiple model teams on shared GPU infrastructure need the ability to enforce per-tenant SLAs — so that one team's training run doesn't starve another team's inference serving. FSx for Lustre has no native multi-tenancy with QoS enforcement. Teams work around this with dedicated file systems per tenant, which multiplies management overhead and cost.

The standard workaround is a dedicated file system per tenant. That solves the isolation problem and creates a new one: every tenant multiplies management overhead, cost, and blast radius. A platform team managing 8 model teams runs 8 separate file systems — each provisioned for peak, each siloed from each other's unused capacity.

WEKA NeuralMesh delivers native multi-tenancy with per-tenant QoS enforcement. Multiple model teams share a single storage cluster with guaranteed throughput floors and ceilings per tenant — no dedicated file systems, no management multiplication, no capacity stranded in a silo.

	FSx for Lustre	WEKA NeuralMesh
Multi-tenancy	No native support; workaround = per-tenant file systems	Native multi-tenant architecture
QoS enforcement	Not enforced; shared-resource contention	Per-tenant throughput floors and ceilings
Management overhead	Multiplies with tenant count	Single cluster, all tenants managed together

Stability AI on AWS: the reference deployment

Stability AI moved from FSx for Lustre to WEKA NeuralMesh on AWS. Same region. Same P4 and P5 instance types. The storage layer changed — so did the economics.

- Storage cost reduced from $5M to $1M annually (95% reduction)
- GPU utilization reached 93% on a 400-node cluster
- Training speed improved 35%
- Autoscale headroom reached 15x on the same AWS footprint
- No migration off AWS — full deployment on existing instance types

WEKA on AWS gave us a 400-node cluster running at 93% GPU utilization with 15x autoscale headroom — on the same AWS footprint. The storage layer was the bottleneck. Replacing it changed our economics.

Stability AI engineering leadership

93%

GPU utilization — Stability AI, 400-node AWS cluster

35%

Faster training on the same instance types

95%

Storage cost reduction, $5M to $1M annually

Read the case study here

Four breaking points, one root cause

Every FSx for Lustre ceiling — throughput coupling, single-AZ, no KV cache, no QoS — traces back to the same architectural decision: FSxL was designed for HPC scratch, not production AI.

- Throughput-capacity coupling: pay for capacity you don't need
- Single-AZ default: one zone failure restarts a multi-day run
- No persistent KV cache: inference TTFT degrades under load
- No QoS: one team's training starves another team's serving

Stop guessing. See your numbers.

Want to benchmark your current setup? Talk to a WEKA expert about your workload, your AWS bill, and where the bottlenecks are. Contact us

Performance results based on Stability AI production deployment on AWS, 2025. Firmus benchmark conducted on NeuralMesh production cluster. Results vary by workload and configuration.