Aug 13, 2025

High-Performance Model Weight Storage and Distribution in Cloud Environments

With the rapid scaling of AI deployments, efficiently storing and distributing model weights across distributed infrastructure has become a critical bottleneck. Here's my analysis of storage solutions optimized specifically for model serving workloads.

The Challenge: Speed at Scale

Model weights need to be loaded quickly during initialization and potentially shared across multiple inference nodes. While local NVMe storage offers blazing-fast speeds of 5-7 Gbps with direct GPU attachment, this approach doesn't scale when you need to:

- Distribute the same model weights to multiple nodes simultaneously
- Update models across a fleet of servers
- Handle dynamic scaling where new nodes need rapid access to model weights

Two Architectural Approaches for Distributed Model Storage

1. NFS-Based Solutions for Model Weights

NFS provides a straightforward path for centralizing model storage. Multiple inference nodes can mount a shared directory containing model weights, enabling:

- Single source of truth for model versions
- Simple model updates (write once, available everywhere)
- POSIX-compliant operations that work seamlessly with existing ML frameworks

2. FUSE-Based Solutions with Intelligent Caching

FUSE implementations can provide smarter model distribution through:

- Lazy loading of model layers (load only what's needed, when it's needed)
- Local caching with intelligent eviction policies
- Tiered storage strategies (hot models in SSD, warm on CDN, cold in object storage)

Scalability

First we will talk about the scalability what we are looking 0 to n machines.

- How do we increase aggregate throughput as demand grows?
- What happens if instead of 1 client 100 clients ask for the data
- How easy is it to scale for fan out workloads

NFS Scaling	FUSE Scaling
Vertical scaling through faster hardware. Horizontal scaling requires complex clustering solutions	Vertical scaling through complex caching mechanisms; virtually unlimited horizontal scale
Performance can degrade with many concurrent clients	Performance scales with parallelization—each client fetches data independently
Single points of failure without proper HA setup	No single points of failure; built-in redundancy and availability per client

Operational Cost

Let's talk about what costs look like for NFS vs Fuse backed storage. What would it means in cost on both operation terms and flexibility

NFS	FUSE backed by Object Storage
Requires dedicated infrastructure and management	Minimal operational overhead – managed service
Higher upfront costs for hardware and setup	No hardware procurement or maintenance
Need for backup and disaster recovery planning	Built-in redundancy and disaster recovery
24/7 operational overhead	Pay-as-you-go operational model

Practical Deployment on CLOUD

There are some overall pointers for considering storage,Let's talk about real world deployment of there storage solutions. What would it cost in terms for deploying this for Major Cloud providers and what can you expect in terms for speed and durability for these solutions in Real world

Let’s talk about solution available across cloud providers for **NFS based storage**:

Cloud	Service	Throughput (MB/s)	Cost (TB/mo)	Protocol	Min Provision
AWS	Amazon EFS (Standard)	50 MB/s 100 MB/s per TiB	$300	NFS v4.0, v4.1	none
AWS	Amazon FSx for Lustre (Persistent SSD)	1024 MB/s per TiB	$980	Lustre (POSIX)	1.2 TiB
AWS	Amazon FSx for NetApp ONTAP	1024 MB/s per TiB	$2200	NFS v3, v4.x	1024 GiB
GCP	Filestore HDD	100 MB/s	$163.84	NFS v3	1 TiB
GCP	Filestore SSD	1200 MiB/s	$300	NFS v3	2.5 TiB
GCP	Cloud NetApp Volumes (Extreme)	1500 MB/s	$399.36	NFS v3, v4.1	1 TiB (pool)
Azure	Azure Files (Premium, Prov v2 SSD)	100-150 MiB/s	$163.84	NFS v4.1	32 GiB
Azure	Azure NetApp Files (Ultra)	1200 MiB/s	$402.17	NFS v3, v4.1	1 TiB (pool)

Real-World Cost Impact for Model Storage (10TB)

Budget Tier: $2,320/month

- Suitable for dev/staging environments
- Handles light concurrent access
- **Max Throughput**: ~1,000 MB/s for 10TB

Performance Tier: $6,920/month

- Production-grade for high-concurrency serving
- 3x the cost, but 10-20x the throughput
- **Max Throughput**: ~10,240 MB/s for 10TB

The buys you the ability to serve hundreds of nodes simultaneously without bottlenecks—often the difference between 5-minute and 30-second model deployment times at scale.

Fuse options that are cloud provider specific

Service	POSIX Compliance	Throughput MB/s	Small File Performance	Large File Performance	Local Cache
AWS Mountpoint-S3	Limited	400-500	Poor	Excellent	Elastic scaling, LRU eviction
Google Cloud	Partial	200-300	Good	Good	Configurable TTL, parallel downloads

Service	POSIX Compliance	Throughput MB/s	Small File Performance	Large File Performance	Local Cache
Storage FUSE					
Azure BlobFuse2	Good	150-250	Very Poor	Moderate	3 modes (Block, File, Streaming)

Fuse options that are Cross Cloud

Provider	Throughput (approx)	License	Cost
cunoFS	~2000 MB/s	Proprietary commercial; free for personal use (registration required), 30 day commercial eval	Contact sales for commercial pricing
JuiceFS	~1000 MB/s reads	Apache 2.0 (Community Edition)	Cloud Service $0.02 / GB / mo; Enterprise – contact sales
Goofys	~500 MB/s max	MIT open-source	Free (open-source)
Alluxio	~1500MB/s depends on RAM/CPU/Net	Apache 2.0 (Core); Enterprise commercial	Open Source: free; Enterprise – contact sales

Real-World Cost Impact for Model Storage (10TB)

Standard Tier: $220/month

- Production-grade for high-concurrency serving
- **Max Throughput**: ~500 MB/s per node (can be tuned for more)

Which is better for ML models weights ?

Cost - Model storage quickly becomes prohibitively expensive with NFS. With modern models ranging from tens of gigabytes to over a terabyte each, and fast NFS solutions charging $500-1,500 per TB monthly. *FUSE cuts storage costs by 95% compared to NFS*, despite both ultimately serving the same purpose: read-only blob distribution.

Performance - NFS's central server architecture becomes a critical bottleneck during scale-out events. Ironically, FUSE-backed object storage achieves 10x better aggregate throughput than 'high-performance' NFS during critical scaling events—when 50 nodes pull models simultaneously, FUSE delivers 25 GB/s total while NFS saturates at 2.5 GB/s, often delivering worse throughput than parallel object storage requests to S3 or GCS.

Getting Start with Cloud-Native FUSE Mounters

Cloud providers offer native FUSE-based solutions that can bridge the gap between object storage economics and NFS-like performance. Here's a practical path to production:

- **AWS**: Use Mountpoint for S3
- **GCP**: Deploy GCSFuse
- **Azure**: Leverage BlobFuse2

Tune for ML Workload Characteristics

- **Page Size**: Increase from default 4KB to 1-2MB to match model file chunk
- **Prefetch Depth**: Configure aggressive read-ahead (256MB+) since model loading is sequential
- **Concurrency**: Set parallel stream counts to 8-12 threads for multi-GB models
- **Cache TTL**: Trigger cache population before pod scheduling to ensure models are locally cached

The Future for Fuse

We need FUSE to evolve from "making object storage barely usable" to "making object storage indistinguishable from local storage" for ML workloads. This means:

1. **Speed**: Matching NVMe performance (5-10 GB/s) through kernel bypass and parallelization.
2. **Compliance**: Supporting every POSIX operation that PyTorch / JAX / TensorFlow might call and use for loading
3. **Intelligence**: Understanding ML access patterns and optimizing for them automatically