NAND RESEARCH

Research Brief:

WEKA NeuralMesh

STEVE MCDOWELL, CHIEF ANALYST

JUNE 2025

WEKA NEURALMESH

WEKA recently announced **NeuralMesh**, its software-defined, microservices-based storage platform engineered for high-performance, distributed AI workloads. NeuralMesh introduces a dynamic mesh architecture that departs from traditional monolithic or appliance-bound storage systems.

It addresses the requirements of emerging AI applications, including agentic AI, large-scale inference, and token warehouse infrastructure, by delivering consistent microsecond-level latency, fault isolation, and scale-out performance across hybrid environments.

NeuralMesh is a strategic shift from WEKA's previous parallel file system and data platform architecture by adopting a fully container-native, service-oriented infrastructure model. This approach enables real-time responsiveness for GPU and TPU pipelines, automated workload optimization, and flexible deployment topologies extending from bare metal to multi-cloud.

UNDER THE HOOD: TECHNICAL DETAILS

At the core of NeuralMesh is a distributed mesh topology built on containerized microservices. Each core storage function, including metadata services, protocol handlers, I/O processing, data protection, and telemetry, runs as an independently orchestrated container.

NeuralMesh borrows several architectural principles from hyperscalers, but it delivers them as a software-defined, deploy-anywhere platform rather than as a cloud-only backend service. It's a model that ensures service-level isolation, independent scalability, and fault containment.

Key architectural elements include:

- **Latency:** NeuralMesh delivers consistent microsecond-level latency by eliminating kernel context switches, bypassing legacy I/O stacks, and leveraging user-space I/O processing. This is achieved through techniques such as zero-copy data paths and RDMA (Remote Direct Memory Access) where supported. These optimizations allow GPUs and TPUs to maintain high utilization during inference and real-time agent execution.
- **Architecture:** The core of NeuralMesh is its dynamic mesh topology. Unlike controller-based or shared-nothing architectures, NeuralMesh enables each node to run independently orchestrated microservices that communicate over a service mesh. This architecture routes data and metadata requests dynamically based on node load, network latency, and service health. The absence of a central metadata server improves scalability and fault tolerance.

- **Elasticity:** Each microservice in NeuralMesh can scale horizontally based on demand. For instance, protocol handlers can be increased independently of replication or telemetry services. This design ensures that bursty AI workloads—such as those common in inference—do not require overprovisioning of the entire stack. Kubernetes-native orchestration enables automatic scale-in and scale-out operations without service interruption.
- **Resiliency:** NeuralMesh exhibits fault-tolerant behavior through self-healing mechanisms. If a node or service fails, the system re-routes operations to healthy nodes and automatically redeploys failed containers. Recovery times are measured in minutes, with no centralized failover points. Distributed metadata and data protection schemes ensure data integrity even in the event of partial network or node failures.
- **Deployment Models:** NeuralMesh supports a wide range of deployment topologies, including on-premises (bare metal), edge locations, public cloud instances, and hybrid/multi-cloud environments. It is fully containerized and integrates with orchestration platforms such as Kubernetes and OpenShift, enabling portable, infrastructure-agnostic deployment. Users can also deploy NeuralMesh as part of a hybrid AI stack that spans both cloud and edge.
- **Integration:** NeuralMesh is API-first and compatible with infrastructure-as-code workflows. It exposes control plane and telemetry interfaces via RESTful APIs, integrates with service mesh tools for observability (e.g., Prometheus, Grafana), and supports CSI drivers for container storage integration. The platform is optimized for AI infrastructure, including compatibility with GPU-aware schedulers and potential alignment with NVIDIA frameworks (e.g., Triton Inference Server, TensorRT).

PERFORMANCE CHARACTERISTICS

NeuralMesh targets performance metrics aligned with the operational demands of real-time inference, multi-agent reasoning, and high-throughput AI training:

- **Latency**: Sustains data access latencies in the low microsecond range, eliminating I/O wait states that often reduce GPU/TPU utilization.
- **Scalability**: Increases aggregate IOPS and throughput as the mesh expands to petabyte and exabyte scales, avoiding the diminishing returns typically associated with traditional shared-nothing or controller-based designs.
- **Utilization Efficiency**: Customers such as Stability AI report GPU utilization increases to 93% during model training, suggesting reduced idle time waiting on storage.

DEPLOYMENT MODELS

NeuralMesh supports flexible deployment scenarios across AI infrastructure stacks:

- **Bare Metal**: For low-level latency and maximum performance in tightly coupled training environments.
- **Cloud and Multicloud**: Native compatibility with container orchestrators enables deployment across public cloud services, including burstable high-performance workloads.
- **Hybrid Edge-Core**: Mesh extensibility supports edge inference workloads with central coordination or data aggregation in core or cloud environments.

Customers can scale deployments incrementally without performing forklift upgrades or rearchitecting their data infrastructure. Upgrades occur with minimal downtime due to containerized service replacement.

BENEFITS VS TRADITIONAL STORAGE

According to the WEKA, NeuralMesh achieves higher efficiency and performance by eliminating architectural chokepoints, decoupling services for granular control, and enabling real-time adaptability.

These capabilities are essential for modern AI infrastructure, where inference throughput, real-time responsiveness, and GPU utilization are critical for both technical success and economic viability.

Attribute	Traditional Storage	NeuralMesh
Architecture	Monolithic or controller-based	Microservices-based
Scaling Model	Static or shared-nothing	Dynamic, self-optimizing mesh
Latency	Milliseconds (typical)	Microseconds (consistent)
Fault Tolerance	Centralized rebuilds, hours	Distributed rebuilds, minutes
Resource Utilization	Static provisioning	Elastic scaling per service
Deployment Flexibility	Appliance-bound, limited cloud support	Cloud-native, edge-ready, multi-cloud compatible
Upgrade Path	Disruptive, hardware-dependent	Rolling, in-place service upgrades

NeuralMesh brings new efficiency and performance benefits over traditional storage through four key architectural innovations:

1. Microservices-Based Architecture

Traditional storage systems are typically monolithic, with tightly coupled services that must scale together and often become bottlenecks. NeuralMesh employs a microservices approach, where every major function—metadata handling, data services, protocol access, telemetry, and replication—is deployed as an independent, containerized service.

Benefits:

- **Elastic scaling** of individual services based on workload demand (e.g., more protocol gateways during inference surges)
- **Service isolation** eliminates cascading failures and noisy neighbor effects
- **Agile updates** and rolling upgrades with minimal system impact

This architectural modularity increases operational efficiency and responsiveness under dynamic AI workloads.

2. Mesh-Based Data Topology

NeuralMesh replaces centralized controllers or shared-nothing architectures with a distributed mesh topology, where each node participates in a self-organizing fabric. Data and metadata requests are routed dynamically across the mesh for optimal performance.

Benefits:

- **No central bottlenecks** (e.g., metadata servers)
- **Parallelism at scale**, with load-aware routing of requests
- **Resilience and self-healing**: nodes fail without impacting system availability or latency

Its mesh model ensures linearly increasing performance as capacity and compute nodes are added, avoiding the performance degradation often seen in legacy scale-out systems.

3. Microsecond-Latency Data Access

Most enterprise storage systems are optimized for throughput (MB/s or GB/s) but not latency (µs), which is critical for inference and agentic AI. NeuralMesh minimizes software overhead and I/O path length by:

- Running in user space, bypassing kernel context switching
- Leveraging zero-copy data paths
- Using RDMA and parallel I/O queues (where supported).

This results in delivering sustained microsecond-level latency even at the petabyte scale, fast enough to keep GPUs and TPUs fully fed and avoid I/O-induced compute stalls.

4. Intelligent, Adaptive Behavior at Scale

NeuralMesh becomes faster and more resilient as it grows, in contrast to legacy systems that degrade under scale. It uses:

- **Decentralized metadata and control services** for improved concurrency
- **Intelligent monitoring and adaptive tuning** to optimize placement, load balancing, and cache utilization in real-time
- **Service-aware routing** that avoids contention points

Additionally, storage-as-code deployment and automation reduce human error and operational drag, leading to higher infrastructure utilization.

WHAT PROBLEM DOES WEKA'S NEURALMESH SOLVE?

NeuralMesh solves the fundamental data infrastructure bottlenecks that limit the scalability, performance, and efficiency of modern AI workloads, particularly real-time inference, agentic AI, and multi-tenant AI factories:

Problem	WEKA's NeuralMesh
Latency bottlenecks for inference	Microsecond data access speeds
Poor GPU utilization	Consistent high-throughput to feed accelerators
Scaling fragility	Self-healing, scale-out mesh architecture
Rigid, monolithic systems	Microservices-based, containerized services
Limited deployment flexibility	Supports bare metal, edge, cloud, and hybrid
Noisy neighbors and contention	Built-in multitenancy with strong isolation
Poor integration with AI/MLOps	Storage-as-code, Kubernetes-native deployment

Let's take a more in-depth look at each of these.

GPU UNDERUTILIZATION DUE TO I/O BOTTLENECKS

AI workloads, especially inference and real-time reasoning, require data to be delivered to accelerators (GPUs/TPUs) with microsecond latency. Traditional storage systems—typically built for throughput rather than ultra-low latency—create data stalls that leave expensive GPUs idle.

This results in:

- Low hardware utilization
- Slower model training or inference
- Higher cloud and energy costs

NeuralMesh addresses this by providing consistent, microsecond-level data access, maximizing GPU/TPU throughput, and enhancing token output efficiency.

FRAGILITY AND INEFFICIENCY AT SCALE

Legacy storage systems often degrade as data and I/O demands grow, particularly at the petabyte-to-exabyte scale. These systems:

- Rely on centralized metadata or controller nodes
- Experience performance bottlenecks under distributed or concurrent access
- Suffer from long rebuild times after node failure

NeuralMesh addresses this with a mesh-based, distributed architecture that becomes faster and more resilient as it scales, offering:

- Parallel metadata and data services
- Self-healing capabilities that recover from failure in minutes
- Scalability without performance tradeoffs

INFLEXIBLE, MONOLITHIC ARCHITECTURES

Most traditional storage platforms are built as monolithic appliances with tightly coupled hardware and software stacks. These rigid designs:

- Require forklift upgrades for scaling

- Limit deployment to fixed environments (on-prem only or single cloud)
- Are difficult to update or reconfigure without downtime

NeuralMesh eliminates these constraints by adopting a microservices-based, container-native architecture that enables:

- Modular scaling of individual services (e.g., protocol handling, telemetry)
- Portability across bare metal, cloud, edge, and multi-cloud
- Seamless upgrades and continuous delivery with minimal disruption

MULTI-TENANCY AND RESOURCE CONTENTION

In shared AI infrastructure environments, such as hyperscale platforms or AI factories, different tenants or workloads often compete for resources. Most legacy systems cannot:

- Isolate workloads cleanly
- Guarantee performance across noisy neighbors
- Offer fine-grained quality of service (QoS)

NeuralMesh provides built-in multitenancy with service-level isolation, resource limits, and QoS enforcement. This ensures predictable performance and fault isolation across diverse AI workloads.

MISMATCH WITH CLOUD-NATIVE AND AI-NATIVE DEVELOPMENT MODELS

Modern AI infrastructure is built on cloud-native, composable services, including Kubernetes, serverless compute, data lakes, and orchestration pipelines. Traditional storage systems don't integrate cleanly into these stacks, which:

- Slows down MLOps and DevOps workflows
- Introduces operational friction and vendor lock-in

NeuralMesh integrates into modern pipelines using:

- API-first design and infrastructure-as-code compatibility
- Declarative deployment models
- Support for container orchestrators and AI development frameworks

USE CASES AND INTEGRATION

NeuralMesh supports several high-performance AI scenarios:

- **Token Warehouses**: For managing large volumes of embedding vectors, retrieval-augmented generation (RAG) datasets, and tokenized model inputs/outputs.
- **Agentic AI**: Supports dynamic and multimodal agents that require fast access to structured and unstructured data across distributed environments.
- **AI Factories**: Centralized pipelines for model training, evaluation, and deployment, where performance scaling and real-time telemetry are critical.
- **Multi-tenant Service Providers**: Hyperscalers and Neocloud platforms serving multiple customers can enforce tenant isolation and performance guarantees using container-level controls.

NeuralMesh integrates with GPU-accelerated platforms and frameworks, including those from NVIDIA, and supports S3-compatible object interfaces to enable workload portability.

COMPETITIVE POSITIONING

WEKA's NeuralMesh introduces a container-native, microservices-based storage architecture targeting inference-era AI infrastructure. It's an approach that offers a differentiated approach for enterprises and service providers building real-time, AI-native infrastructure, particularly where latency, elasticity, and GPU efficiency are critical. I

Its primary distinction lies in bringing hyperscale-like storage primitives to organizations outside the hyperscaler tier.

This contrasts with the evolutionary approaches taken by traditional storage companies, such as Dell Technologies, NetApp, and Pure Storage. It also offers strong differentiation from more adventurous approaches, such as VAST Data's recently announced hyper-converged AI solution (HCAI), VAST AI OS,

This section is only available to NAND Research clients and IT Advisory Members. Please reach out to info@nand-research.com to learn more.

ANALYSIS

The launch of NeuralMesh marks WEKA's entry into the emerging category of AI-native storage infrastructure. While WEKA previously led in high-performance file systems for machine learning, NeuralMesh elevates its architecture to directly address real-time inference and agentic AI demands.

WEKA is positioning NeuralMesh to address emerging challenges in the inference-era infrastructure. Its emphasis on microsecond latency, software-defined deployment, and elastic service orchestration differentiates it from traditional HPC or appliance-centric storage. While WEKA has an established presence in AI training, NeuralMesh extends its reach to real-time, production-grade AI use cases.

NeuralMesh also aligns WEKA more closely with modern DevOps and MLOps practices, positioning the platform as an infrastructure-native component in AI factories. Its support for cloud-native orchestration and its scalability model allow for integration into next-generation AI pipelines without vendor lock-in or infrastructure sprawl.

This puts competitive pressure on vendors with appliance-centric or controller-based architectures that lack the elasticity, service modularity, or microsecond responsiveness required for modern AI infrastructure.

NeuralMesh is a strong, deliberate pivot by WEKA toward the next phase of AI infrastructure, one where inference, agentic AI, and real-time responsiveness drive platform requirements. Its dynamic mesh architecture and microservices-based model offer technical differentiation.

As AI production workloads evolve, NeuralMesh provides a credible and forward-looking option for organizations building AI factories or seeking to operationalize inference at scale. It’s a bold move, but one that brings much-needed new efficiencies to enterprise AI workflows.

The information contained within this document was believed by NAND Research to be reliable and is provided for informational purposes only. The content may contain technical inaccuracies, omissions, or typographical errors. This document reflects the opinions of NAND Research, which is subject to change. NAND Research does not warranty or otherwise guarantee the accuracy of the information contained within.

NAND Research is a technology-focused industry analyst firm providing research, customer content, market and competitive intelligence, and custom deliverables to technology vendors, investors, and end-customer IT organizations.

Contact NAND Research via email at info@nand-research.com or visit our website at nand-research.com.