Synapse: High-Performance Federated Learning

Dr. Sarah Jenkins Prof. Thomas Patel

Meridian Research Labs

Presentation Outline

Part I: Architecture & Design

1. Distributed Orchestration
2. Synapse System Topology
3. Optimization Framework

Part II: Performance & Security

4. Hardware Benchmarking
5. Privacy-Preserving Mechanics
6. Deployment Roadmap

Part I: Architecture & Design

Distributed Orchestration on Vertex Edge Nodes

Orchestration Challenges in Edge AI

Goal: Train a unified model across decentralized data resources without raw upload.

- **Communication Latency:** High-dimensional updates saturate shared channels.

Orchestration Challenges in Edge AI

Goal: Train a unified model across decentralized data resources without raw upload.

- **Communication Latency:** High-dimensional updates saturate shared channels.
- **Resource Asymmetry:** Vertex nodes span low-power IoT to compute clusters.

Orchestration Challenges in Edge AI

Goal: Train a unified model across decentralized data resources without raw upload.

- **Communication Latency:** High-dimensional updates saturate shared channels.
- **Resource Asymmetry:** Vertex nodes span low-power IoT to compute clusters.
- **Statistical Variance:** Non-IID local data distributions delay convergence.

System Topology Paradigm

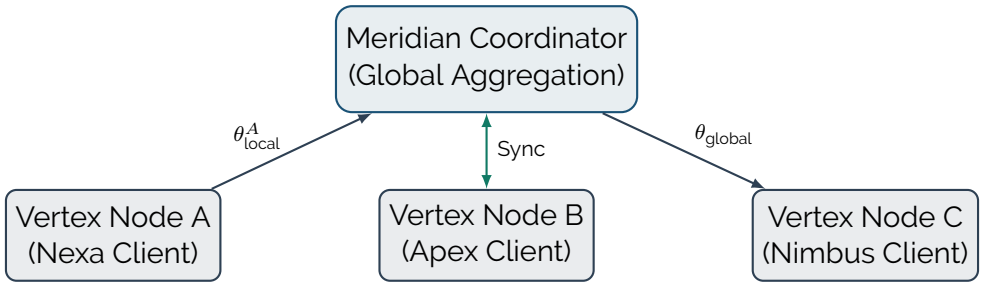
Traditional Centralized Training

- Direct raw upload to centralized database.
- Saturated uplink networks during peak times.
- Single-point-of-failure under breach.

Synapse Federated Protocol

- Local optimization on local device storage.
- Only parameterized weights are transmitted.
- Data remains isolated inside the local edge.

Synapse Optimization Loop



Interactive aggregation loop between Meridian Coordinator and local clients.

Optimization Objective

We formulate the distributed empirical risk minimization as:

$$\min_{\theta \in \mathbb{R}^d} F(\theta) = \sum_{k=1}^{K} p_k F_k(\theta)$$

where the local loss function $F_k(\theta)$ for client k is computed over dataset $\mathcal{D}_k$:

$$F_k(\theta) = \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} \ell(f(x_i; \theta), y_i)$$

Here, client weights are proportional to dataset sizes: $p_k = |\mathcal{D}_k| / \sum_j |\mathcal{D}_j|$.

Part II: Performance & Security

Differential Privacy and Security Benchmarks

Heterogeneous Device Performance

Comparative evaluations of Synapse training runs on various client edge nodes.

Platform	Node Type	Latency (s)	Accuracy (%)
Apex	Compute Heavy	12.4	94.2
Vertex	Standard Edge	24.8	93.8
Nexa	Low-Power IoT	85.2	91.5
Nimbus	High-Performance	8.9	94.5

Training metrics compiled over 100 federated epochs.

Synapse Security Architecture

Differential Privacy

Strict bounding of local update sensitivity to protect user membership information.

Local Perturbation

Additive Gaussian noise is parameterized per-epoch based on privacy budget constraints.

Aggregator Threat Model

The coordinator is considered curious but honest. Synapse deploys cryptographic secret sharing to prevent weight snooping.

Milestones & Future Roadmap

- **Secure Aggregation Expansion (Q4 2026):** Deploying multi-party computation across all Apex and Nexa standard nodes.
- **Asynchronous Updates (Q2 2027):** Alleviating straggler issues on low-bandwidth IoT devices.
- **Hardware Enclaves (Q4 2027):** Leveraging Vertex trusted execution environments (TEEs) for isolated training.

Resources & Contacts

Additional developer tools and contact details for the Synapse project.

- Project Homepage: `https://meridian-labs.org/synapse`
- Security disclosures: `https://meridian-labs.org/sec`
- Scheduled presentation: NeurIPS 2026
- Developer Lab Hotline: +1-555-0199

Thank You!

Questions & Discussion

Meridian Research Labs · Synapse Project