Scalable Decentralized Optimization for Large Language Models

Introducing the Vertex-SGD Framework

Dr. Sarah Vance Prof. Thomas Miller

Vertex AI Research Lab, Synapse Technologies

August 2, 2026

Outline

Challenges in Large-Scale LLM Training

- **High Communication Overhead**: Dense parameters result in all-reduce bottlenecks on large clusters.

Challenges in Large-Scale LLM Training

- **High Communication Overhead**: Dense parameters result in all-reduce bottlenecks on large clusters.
- **Single-Point Failure**: Centralized parameter server topologies pose massive reliability risks.

Challenges in Large-Scale LLM Training

- **High Communication Overhead**: Dense parameters result in all-reduce bottlenecks on large clusters.
- **Single-Point Failure**: Centralized parameter server topologies pose massive reliability risks.
- **Resource Underutilization**: Asymmetric compute nodes lead to idle wait states during barrier synchronization.

Challenges in Large-Scale LLM Training

- **High Communication Overhead**: Dense parameters result in all-reduce bottlenecks on large clusters.
- **Single-Point Failure**: Centralized parameter server topologies pose massive reliability risks.
- **Resource Underutilization**: Asymmetric compute nodes lead to idle wait states during barrier synchronization.
- **Scalability Limits**: Traditional SGD efficiency drops drastically beyond a few dozen nodes.

Centralized vs. Decentralized Optimization

Centralized Paradigm

- Rely on dedicated parameter servers.
- High bandwidth required at the root.
- Synchronous gradient aggregation.
- Vulnerable to straggler nodes.

Decentralized (Vertex-SGD)

- Local consensus updates via neighbors.
- Uniform network traffic distribution.
- Asynchronous execution models.
- High robustness to node failures.

The Vertex-SGD Optimization Protocol

Let $x_i^{(t)}$ represent the local parameter weights of node i at step t.
The local update consists of a gradient step followed by neighbor consensus:

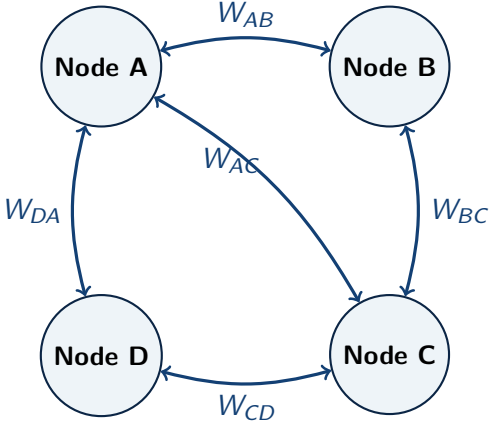
$$x_i^{(t+1/2)} = x_i^{(t)} - \eta g_i \left(x_i^{(t)} \right) \tag{1}$$

$$x_i^{(t+1)} = \sum_{j \in \mathcal{N}_i} W_{ij} x_j^{(t+1/2)} \tag{2}$$

- $\eta > 0$ is the local learning rate.
- g_i is the stochastic gradient computed on node i's local batch.
- $W \in \mathbb{R}^{n \times n}$ is a doubly stochastic gossip matrix.
- $\mathcal{N}_i$ is the set of physical neighbors of node i.

Communication Topology & Adjacency

The network graph determines the speed of consensus convergence.



Experimental Configurations

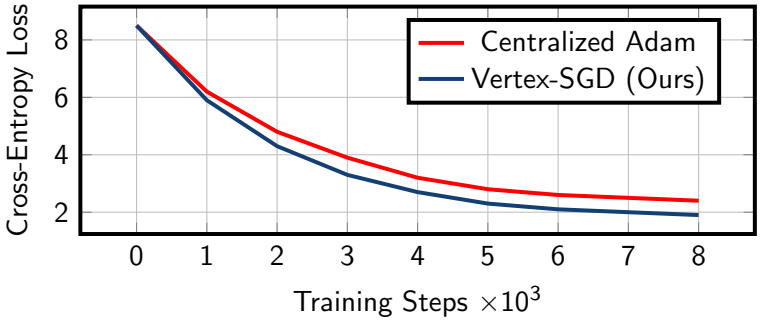
We evaluate Vertex-SGD on the custom Apex-8B model cluster:

Table: Comparison of training settings on Vertex clusters

Cluster Configuration	Optimization Type	Avg. Bandwidth (GB/s)	Step Time (ms)
Vertex-32 (32 nodes)	Centralized Adam	100	420
Vertex-32 (32 nodes)	Vertex-SGD (Ours)	12	180
Vertex-128 (128 nodes)	Centralized Adam	400	890
Vertex-128 (128 nodes)	Vertex-SGD (Ours)	15	210

Convergence Efficiency

Vertex-SGD maintains the convergence quality of centralized methods while decreasing wall-clock time.



Key Observations & Robustness

Theoretical Convergence

Local parameters approach the true global optimum sublinearly for non-convex loss surfaces.

Straggler Vulnerability

Strict consensus bounds might delay updates if individual node computation speeds degrade.

Empirical Speedup

Over a wide range of tasks, Vertex-SGD yields a $2.1\times$ to $4.2\times$ reduction in end-to-end wall-clock training times.

Future Work & Outlook

We plan to extend Vertex-SGD to address the following challenges:

- **Asynchronous Updates**: Relaxing consensus synchrony to allow fully lock-free training.
- **Adaptive Topologies**: Modifying matrix weights dynamically based on live throughput metrics.
- **Quantized Gossip**: Lowering communication bandwidth via lossy gradient quantization.

Thank You!

Questions & Discussion

sarah.vance@vertexai.org
vertexai.org/research
+1 (555) 019-9283