Review and Critical Analysis

Performance Analysis and Comparison of Distributed Machine Learning Systems

Alqahtani, S. & Demirbas, M. (arXiv:1909.02061, 2019)
https://arxiv.org/abs/1909.02061

1 Summary

This paper presents a systematic analytical and empirical study of communication bottlenecks in distributed Deep Neural Network (DNN) training. Building upon their previous comparative work (2017), Alqahtani and Demirbas focus here on the underlying *communication topologies*: **Parameter Server (PS)**, **Peer-to-Peer (P2P)**, and **Ring Allreduce (RA)**.

The authors develop mathematical cost models for training latency that explicitly account for network bandwidth contention. By validating these models against experiments using TensorFlow and Horovod on Amazon EC2, they demonstrate that the Ring Allreduce architecture offers superior scalability. The study concludes that RA succeeds by decoupling network usage from the number of workers and effectively overlapping computation with communication, whereas PS architectures suffer from severe bandwidth saturation at the central server node.

2 Architectural Modeling

The core contribution of the paper is the formalization of latency models T_{total} for one training epoch, decomposed into computation time ($T_{processing}$) and communication time (T_{tcp}).

2.1 Parameter Server (PS)

The PS architecture employs a central authority (or sharded group of authorities) to maintain global state. Workers push gradients and pull updated weights.

- **Bottleneck Analysis:** The authors model the available bandwidth ($available_B$) for a worker as inversely proportional to the total number of workers w competing for the server's interface (Equation 8):

$$available_B = \frac{\text{TotalBandwidth} \times N_{ps}}{w} \tag{1}$$

- **Latency Model:** Consequently, the communication time grows linearly with the cluster size. The push/pull operations are modeled as dependent on the model size W:

$$T_{tcp-ps} \propto \text{epoch} \times \frac{W}{available_B} \tag{2}$$

  This formulation mathematically predicts the congestion collapse observed in experiments when w increases.

2.2 Peer-to-Peer (P2P)

In this study, P2P is defined as a topology where worker and server processes coexist on the same machine, distributing the parameter shards across the cluster (effectively a distributed PS).

- **Bottleneck Analysis:** While this eliminates the single central bottleneck, the "all-to-all" communication pattern still results in network contention. The model shows that bandwidth is shared among $2(w-1)$ active links.
- **Outcome:** The paper finds P2P performs better than a single PS node but still suffers from load imbalances due to varying tensor sizes in different model layers.

2.3 Ring Allreduce (RA)

Based on the Baidu Allreduce algorithm and adapted by Uber's Horovod, RA organizes workers in a logical ring.

- **Bandwidth Optimality:** Data flows only between neighbors ($i \to i+1$). The amount of data sent by each node is $2\frac{W}{w}(w-1)$, which converges to $2W$ as $w \to \infty$. Crucially, the bandwidth usage per link is *constant*, independent of the cluster size w (Equation 20):

$$T_{tcp-ring} = \frac{2(w-1)\frac{W}{w}}{\text{Bandwidth}} \approx \frac{2W}{\text{Bandwidth}} \tag{3}$$

- **Pipelining:** The architecture naturally facilitates Tensor Fusion, where gradient computation at lower layers overlaps with the transmission of gradients from higher layers (which are computed first during backpropagation).

3 Experimental Evaluation

3.1 Setup

- **Hardware:** Amazon EC2 m4.xlarge instances (4 vCPU, 16GB RAM). *Note: The experiments rely on CPUs, which influences the computation-to-communication ratio.*
- **Software:** TensorFlow v1.11 and Horovod (MPI-based).
- **Workload:** MNIST Image Classification using a Multi-Layer Perceptron (MLP) with two hidden layers.
- **Metrics:** Throughput (images/sec) and Latency (seconds/epoch).

3.2 Key Findings

1. **Scalability of RA:** The Ring Allreduce architecture achieved near-linear scaling. The throughput increased linearly with the number of workers (Fig. 18), and epoch time decreased sub-linearly, validating the constant bandwidth consumption model.
2. **Collapse of PS:** The 1-PS setup degraded rapidly. Throughput peaked at roughly 5 workers and then dropped due to CPU and network saturation at the server (Fig. 6). Adding more PS nodes (2PS, 4PS) alleviated this but introduced coordination overhead.
3. **Computation/Communication Overlap:** The RA system demonstrated high efficiency in hiding communication latency behind computation, a feature explicitly highlighted as a deficiency in standard PS implementations of that era.

4 Critical Analysis

4.1 Strengths

- **Analytical Rigor:** The paper moves beyond "black-box" benchmarking by providing closed-form equations for communication costs. This allows for theoretical predictions of scalability limits before deployment.
- **Topological Clarity:** The distinction between P2P (co-located sharding) and Ring Allreduce is often blurred in literature; this paper clearly delineates them and explains why RA is bandwidth-optimal.
- **Validation of Horovod:** The paper serves as an independent validation of the efficiency of the Horovod framework, which was gaining significant traction at the time (2018-2019) as the standard for distributed TensorFlow.

4.2 Limitations

- **Workload Simplicity (MNIST on CPU):** The choice of MNIST (small model, small images) and CPU-based training is the study's most significant limitation.
  - **Compute/Comm Ratio:** Deep learning on CPUs is compute-bound compared to GPUs. On high-end GPUs, the computation time $T_{processing}$ shrinks drastically, making the communication bottleneck T_{tcp} even more pronounced than reported here.
  - **Model Size:** The small MLP model does not stress the bandwidth as much as modern Large Language Models (LLMs) or ResNets, potentially masking some synchronization latency issues in RA.
- **Network Hardware:** The experiments use standard EC2 networking (likely 1Gbps for m4.xlarge). In modern HPC clusters with 100Gbps InfiniBand or NVLink, the trade-offs between PS and RA shift, particularly regarding latency vs. bandwidth.

5 Conclusion

Alqahtani and Demirbas provide a crucial piece of the puzzle in understanding why the Deep Learning community shifted from the Parameter Server model (2012-2016) to the Ring Allreduce model (2017-Present) for synchronous SGD. By mathematically proving the bandwidth optimality of RA and demonstrating the congestion collapse of centralized PS architectures, the paper offers a solid theoretical foundation for system architects. While the experimental workload is lightweight by modern standards, the architectural insights regarding bandwidth saturation and topology remain valid for large-scale cluster design.