ExaComm'24 Panel

Murali Emani
Argonne Leadership Computing Facility
memani@anl.gov

Evaluate Communication Overhead with Distributed LLM Training

Larger LLM models require distributed implementations: data, model (tensor, pipeline)-parallelism

Among all the ZeRO stages, as the model parameters increase, the communication volume also increases, leading to a higher communication distribution during a training iteration.

This reduces the non-overlapping computation time and makes the overall training throughput further bottlenecked by the network bandwidth

A bottom-up analysis of compute and communication time estimation, as well as the scaling effects across varying numbers of nodes and sharding strategies

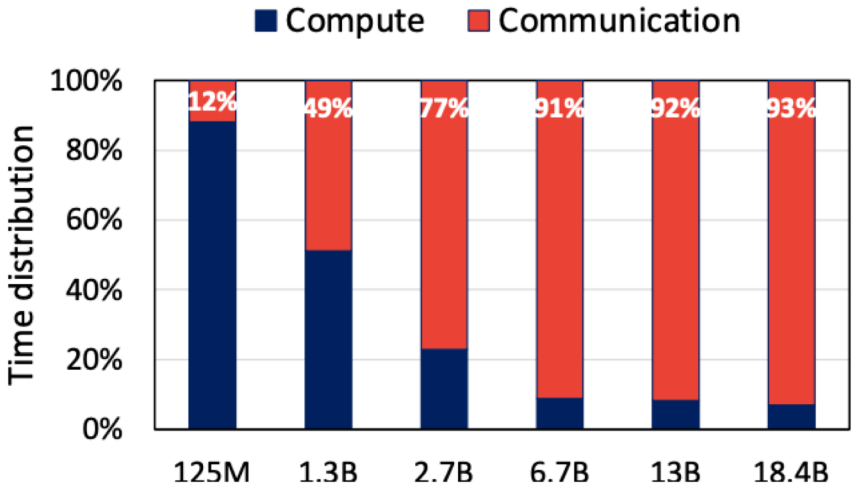


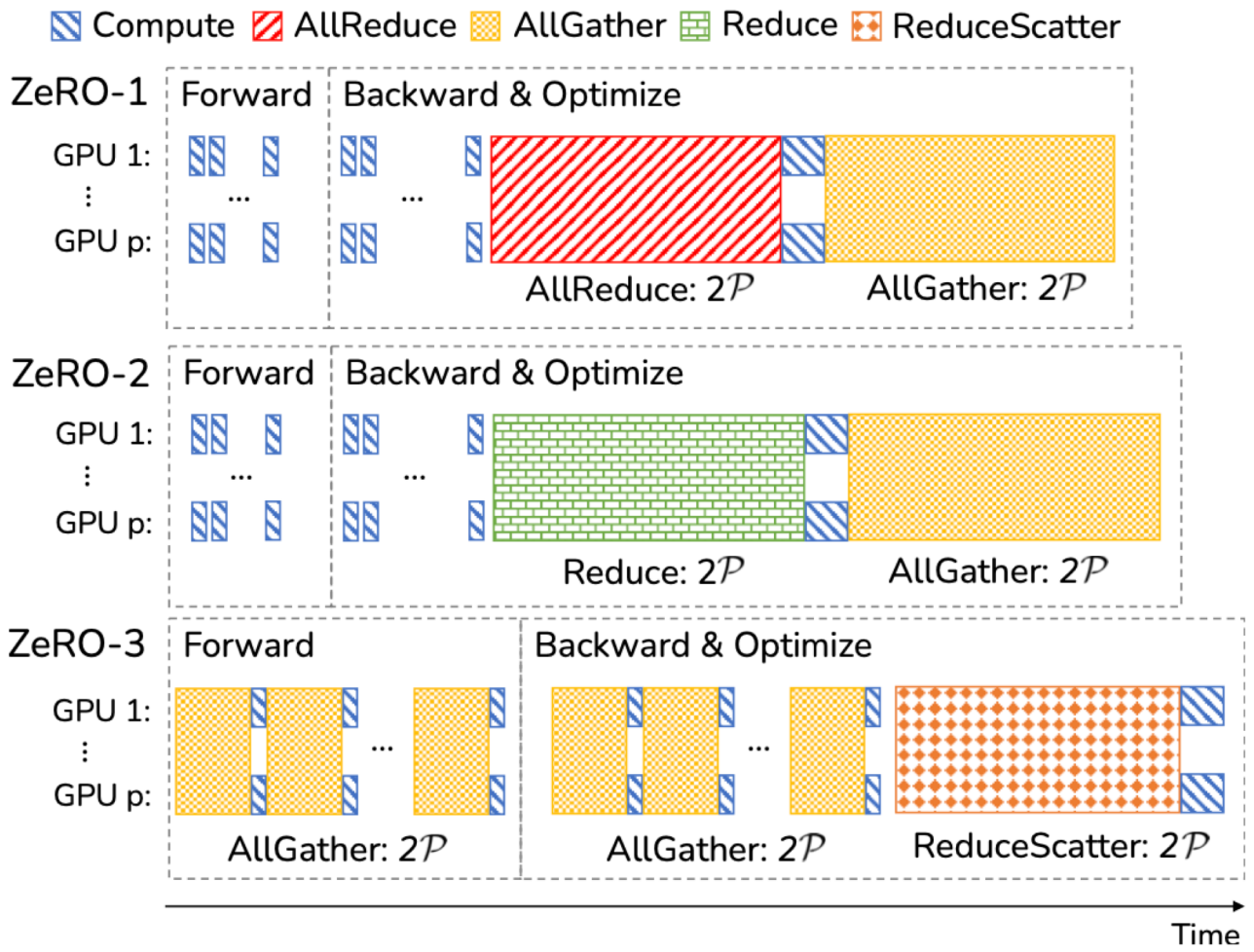
Fig. 2. Time distribution of compute and communication per training iteration for increasing transformer model sizes. [2]

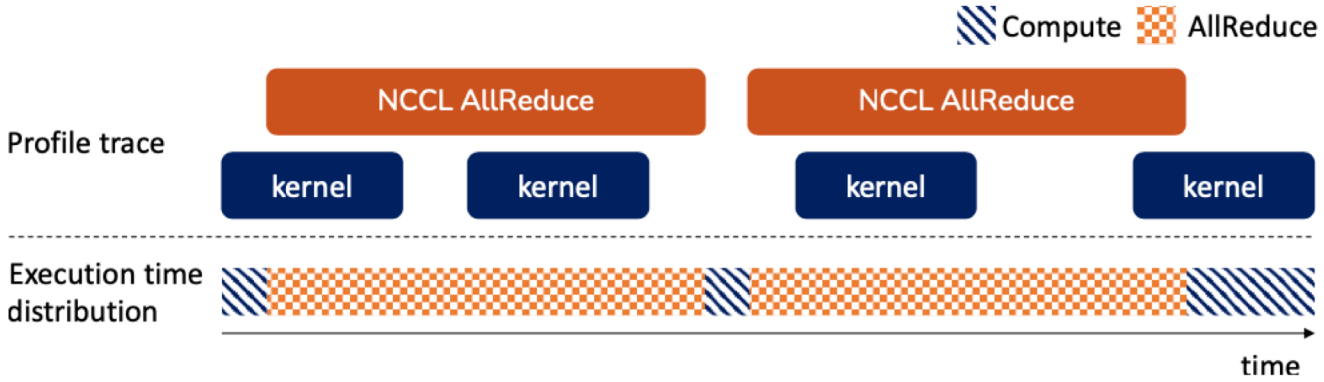
[ACM SIGMETRICS'24] Thorough Characterization and Analysis of Large Transformer Model Training At-Scale,
Scott Cheng, Jun-Liang Lin, Murali Emani, Siddhisanket Raskar, Sam Foreman, Zhen Xie, Venkatram Vishwanath, Mahmut Taylan Kandemir

- Evaluate end-to-end transformer training on
  - — both high-bandwidth (DGX) and low-bandwidth systems upto 512 GPUs,
  - — six model sizes up to 18.4B,
  - — 3 data parallel sharding strategies (ZeRO stages) and 3 combinations of model parallelism,
  - —2 model architectures (GPT and BERT) that involve 4 kinds of collective communication calls
  - —three emulated limited network bandwidth cases in model parallel scaling.

- Our performance modeling is applicable to different model sizes and system architectures.
- Project the speedup at large-scale training runs, achieving mean squared error of less than 2.0%, compared to our evaluation results.

Collective Communication ops in ZeRO



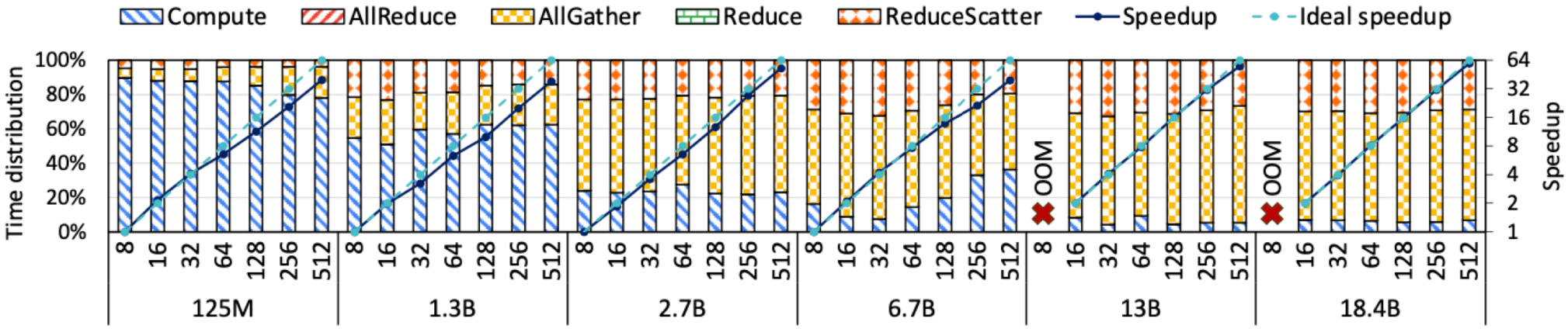


(a) Converting a profile trace with overlapping compute and communication kernels into the execution time distribution.

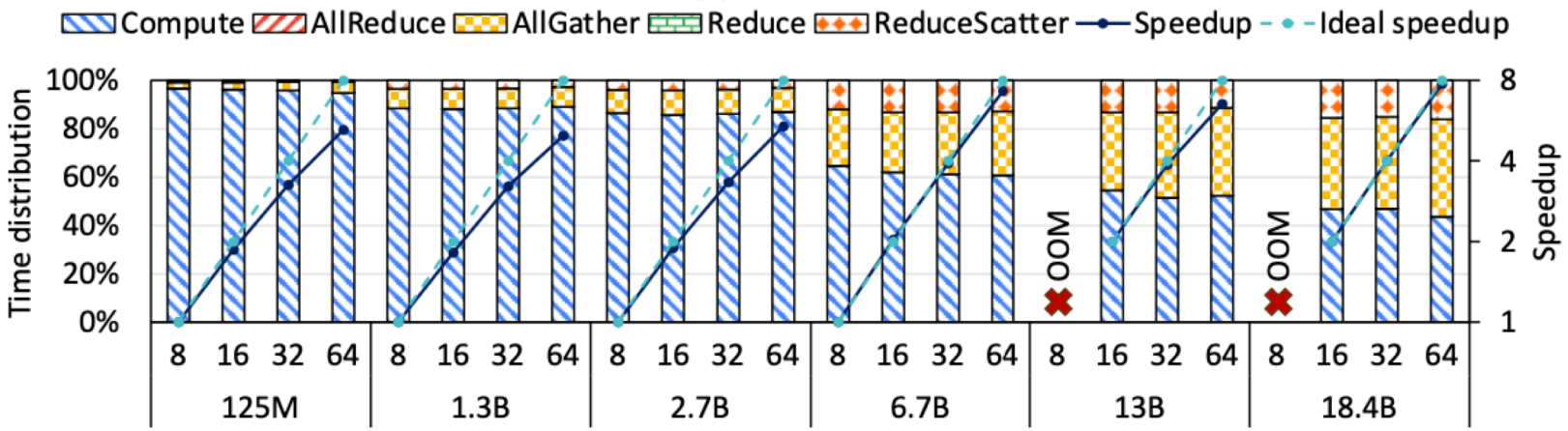
$$\text{speedup}_{g\to n} = \frac{t^{(g)}_{\text{iter.}}}{t^{(n)}_{\text{iter.}}} = \frac{t^{(g)}_{\text{compute}} + t^{(g)}_{\text{comm.}}}{t^{(n)}_{\text{compute}} + t^{(n)}_{\text{comm.}}} = \frac{r^{(g)}_{\text{compute}} + r^{(g)}_{\text{comm.}}}{r^{(g)}_{\text{compute}} + r^{(g)}_{\text{comm.}}/\text{eff}_{g\to n}}.$$

	Polaris	TG40[5]	TG80[5]
System	HPE Apollo	NVIDIA DGX	NVIDIA DGX
Nodes	560	22	2
#Nodes (#GPU) scaled	128 (512)	8 (64)	2 (16)
CPU Model	AMD 7543P	AMD 7742	AMD 7742
CPU Socket(s)	1	2	2
GPU	NVIDIA A100	NVIDIA A100	NVIDIA A100
per GPU Memory	40GB HBM2	40GB HBM2	80GB HBM2e
#GPU per node	4	8	8
GPU Memory B/W	1555 GB/s	1555 GB/s	2039 GB/s
#NVLink per GPU	12 (4 per peer)	12 (NVSwitch)	12 (NVSwitch)
Compute NIC	ConnectX-5	ConnectX-6	ConnectX-6
#Interconnect per node	2	8	8
Total NIC B/W per node	200 Gbps (25 GB/s)	1.6 Tbps (200 GB/s)	1.6 Tbps (200 GB/s)
pinned memory copy B/W	24.6 GB/s	26.1 GB/s	26.2 GB/s
pageable memory copy B/W	19.2 GB/s	12.2 GB/s	12.4 GB/s
P2P B/W	80.5 GB/s	277.6 GB/s	278.8 GB/s

Table 4. The GPU supercomputing systems evaluated in this study.



(a) Polaris.



(b) TG40.

Fig. 14. End-to-end per-iteration training time distribution and speedup for different number of GPUs, model parameters using ZeRO-3 stage.

- Our evaluation results for data parallelism training indicate that a lower bandwidth system may result in
  - 7.35x increase in communication time
  - 59.25% reduction in training throughput

- This impact becomes much more significant in ZeRO-2, where lower network efficiency leads to even worse scaling on the lower bandwidth system with more GPUs.

Key Takeaways

- For future scaling analyses on large-scale systems, it is recommended to first obtain a breakdown of single-node communication and compute times, along with network efficiency through benchmarking.

- With this information, we can estimate end-to-end training times after system scaling

- In a bandwidth-limited environment, network efficiency remains a critical indicator for multi-node scaling, and low-precision compression can help reduce the volume of communication.

Key Takeaways

- In large transformer model training, data parallelism (DP) can achieve better scaling compared to tensor parallelism.
  - — DP parallelism reduces the gradient of weights, while TP reduces the gradient of activations, which are typically much larger than the weight

- For model parallelism, we conclude that
  - —the degree of pipeline parallelism should be larger than the number of nodes
  - —degree of tensor parallelism should not exceed the number of GPUs per node.

- To further improve GPU utilization, we can choose the maximum micro-batch size that can fit into a single GPU memory, and the corresponding global batch size can be determined.

Focus Areas

- Memory stacked memory
- Disaggregated memory
- Algorithms to make best use of this hierarchy

- Optimizations from model implementation, frameworks
- Better collective communication libraries

- N/W bandwidth is critical for emerging workloads at scale
- Manage interference in multi-tenancy latency
- Focus on overlap communication with computation
- Optical interconnects is interesting!

Thank You

- This research was funded in part and used resources of the Argonne Leadership Computing Facility (ALCF), a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Please reach out for further details
Murali Emani, memani@anl.gov