Efficient Allreduce Algorithm for Large-Scale Deep Learning on Distributed Loop Networks

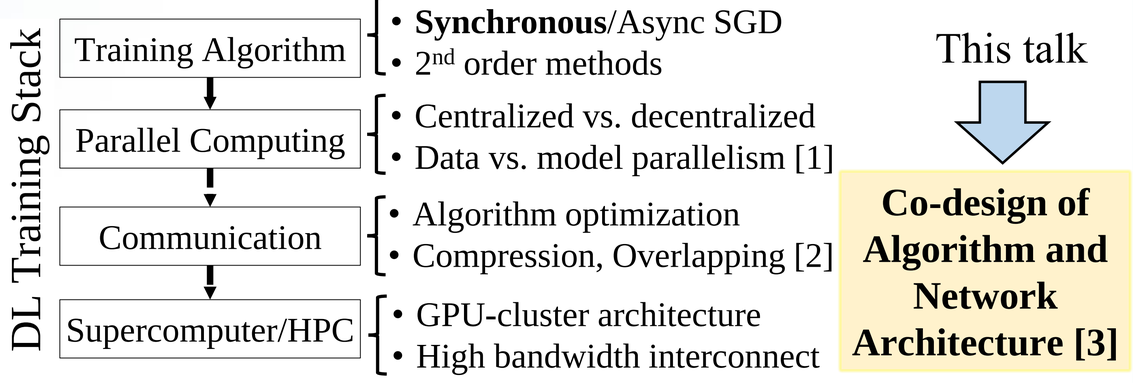
Truong Thao Nguyen, Peng Chen, Yusuke Tanimura
National Institute of Advanced Industrial Science and Technology (AIST),Tokyo, Japan

Research Goal (What?)

Driven by the increase in complexity and size of Deep Learning models, training models on a large-scale computer system is becoming a commonplace. When the number of computing nodes has significantly increased, e.g., **1,000s of GPUs** (large scale), **communicating millions-billions gradients** at each iteration becomes a crucial bottleneck. In this work, we target on speeding up the training phase of Large-Scale Deep Learning on GPU-Cluster.

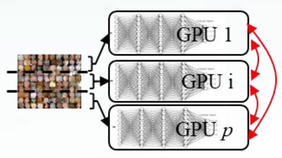
How?

We aim at answering several key questions including:



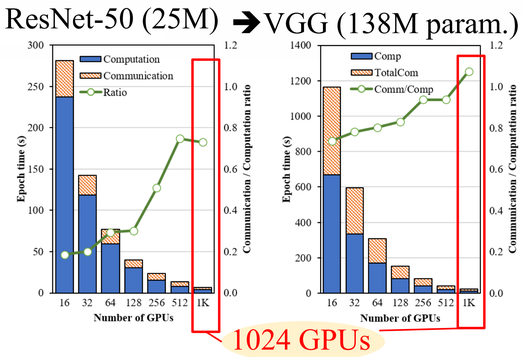
Background

1. Data Parallelism becomes practical



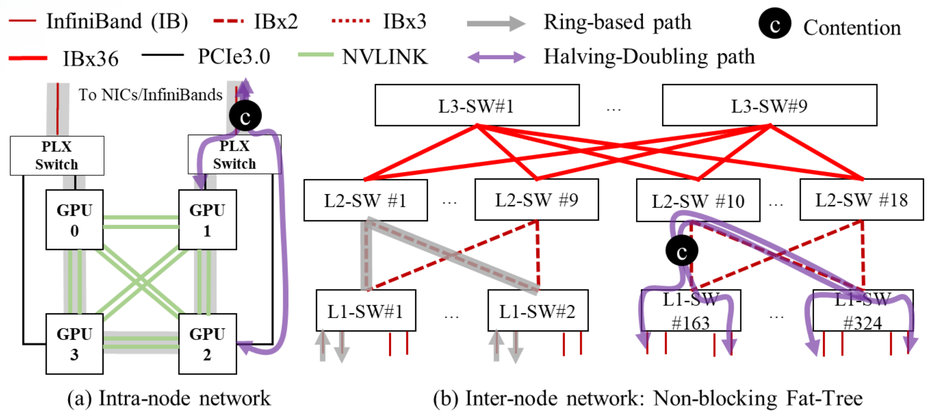
- Communication becomes bottleneck

ResNet-50 (25M) ➔VGG (138M param.) ➔



SOTA 10^{12} param. ?

2. Inter-node communication becomes bottleneck



(a) Intra-node network (b) Inter-node network: Non-blocking Fat-Tree

Ring-based algorithm (NCCL2.x)

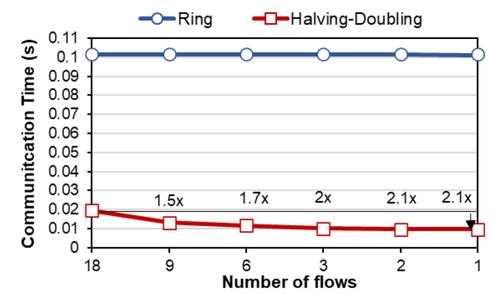
$$2(P-1)\,\alpha_{inter} + \frac{2(P-1)N}{P}\beta_{inter}$$

Halving-Doubling (HD) algorithm

$$2log(P)\,\alpha_{inter} + \frac{2(P-1)N}{P}\beta_{inter}$$

3. Network Contention Problem

- Ring-based: No contention
- HD-based: YES
  - 2x bandwidth reduction

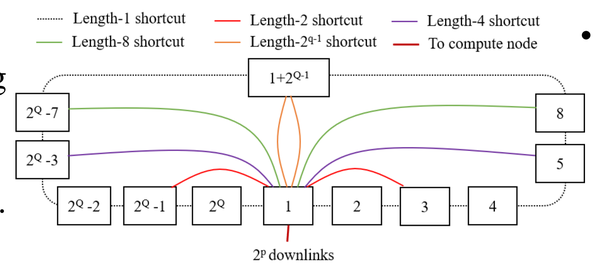


How to apply HD algorithms effectively?

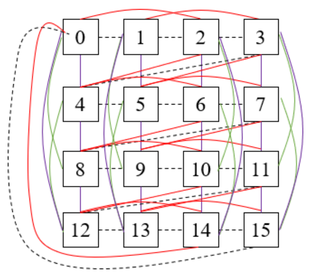
Co-design of Algorithm and Network Architecture

A.1 Logical network topologies (DLN(q,p) [3])

- 2^q switches arranged into a ring with shortcuts connect a switch i to the switch $i + 2^k$.

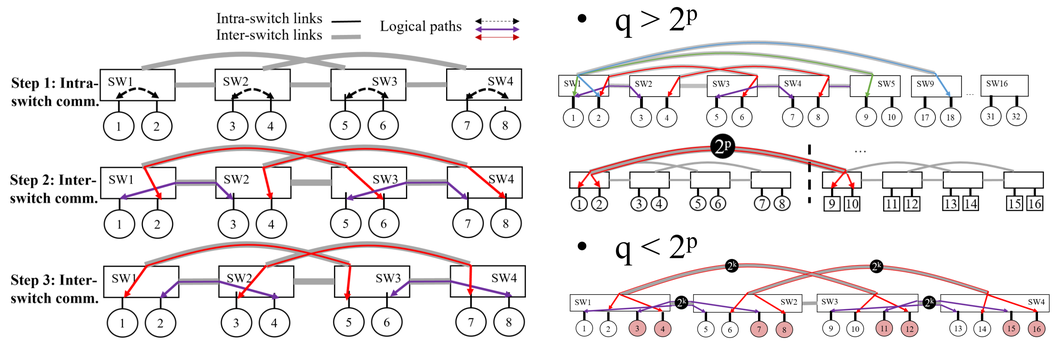


- Difficult to deploy in a server room (cost)



A.2. Communication Algorithm - Shifted Halving-Doubling (SHD)

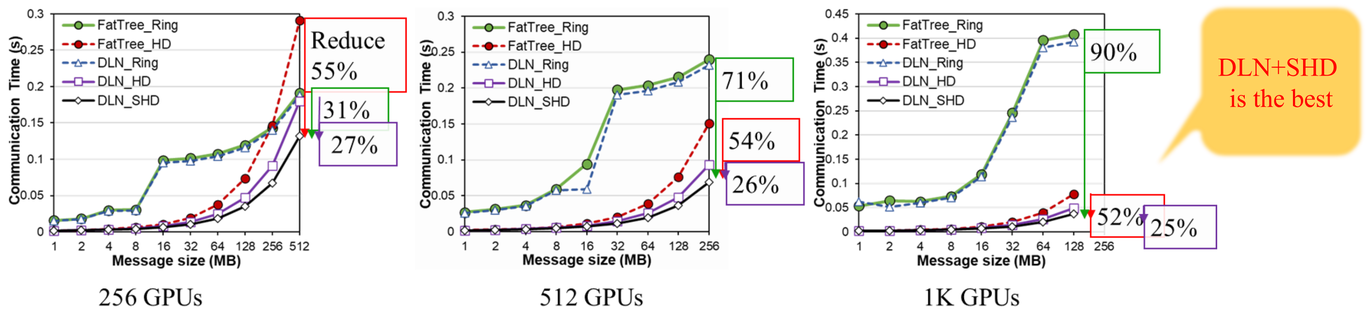
- Fully utilizes all inter-switch links at each step
- Ideal case (q= 2^p): At step j^{th}, PE i^{th} of a switch u is paired with the PE i^{th} of switch $u + 2^{(i+j) \bmod q}$



- $q > 2^p$

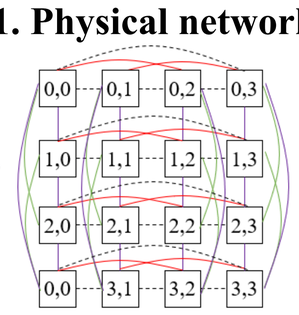
- $q < 2^p$

A.3. Simgrid Simulation Allreduce benchmark



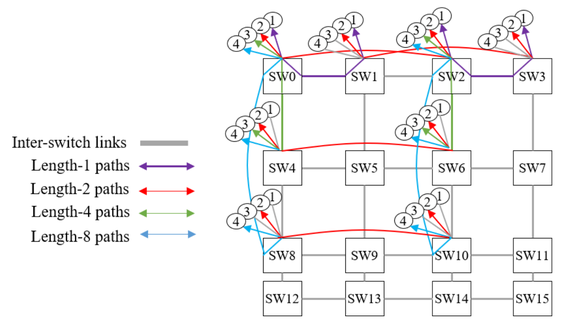
256 GPUs 512 GPUs 1K GPUs

B.1. Physical network topologies - 2D DLN(x,y,z)



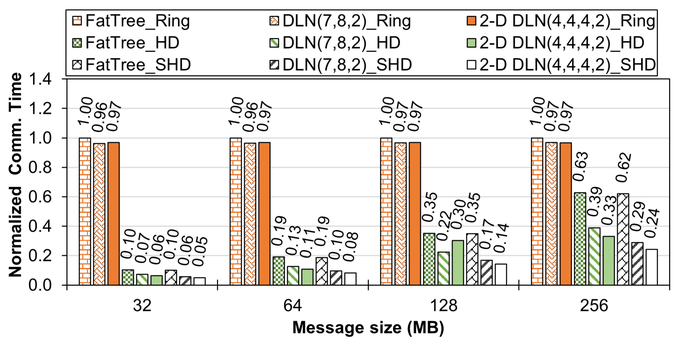
- Arranges switches into a grid of $2^x \times 2^y$ switches with z down link/switch connected to the PEs.
- Map exactly to cabinet → no diagonal cables

B.2. Extended



- SHD still work with 2-D DLN

B.3. Simulation result with 512 GPUs



Conclusion

1. Our co-design DLN + SHD shows better performance than the recent combination of network and algorithms.
2. 2D-DSN help to solve the network implementation issue while do not require changing much in algorithm.
   - A slightly improvement of performance.

References

[1] Kahira, Albert Njoroge, **Truong Thao Nguyen**, Leonardo Bautista Gomez, Ryousei Takano, Rosa M. Badia, and Mohamed Wahib. "An oracle for guiding large-scale model/hybrid parallel training of convolutional neural networks." In Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing, pp. 161-173. 2021.

[2] **Truong Thao Nguyen**, Mohamed Wahib, and Ryousei Takano. "Efficient MPI-AllReduce for large-scale deep learning on GPU-clusters." *Concurrency and Computation: Practice and Experience* 33, no. 12 (2021): e5574.

[3] **Truong Thao Nguyen**, and Mohamed Wahib. "An allreduce algorithm and network co-design for large-scale training of distributed deep learning." In 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 396-405. IEEE, 2021.