The 8th Asia-Pacific Workshop on Networking (APNet 2024)

Understanding Communication Characteristics of Distributed Training

Wenxue Li[1], Xiangzhou Liu[1], Yuxuan Li[1], Yilun Jin[1], Han Tian[2], Zhizhen Zhong[3], Guyue Liu[4], Ying Zhang[5], Kai Chen[1]

[1]iSING Lab, Hong Kong University of Science and Technology,

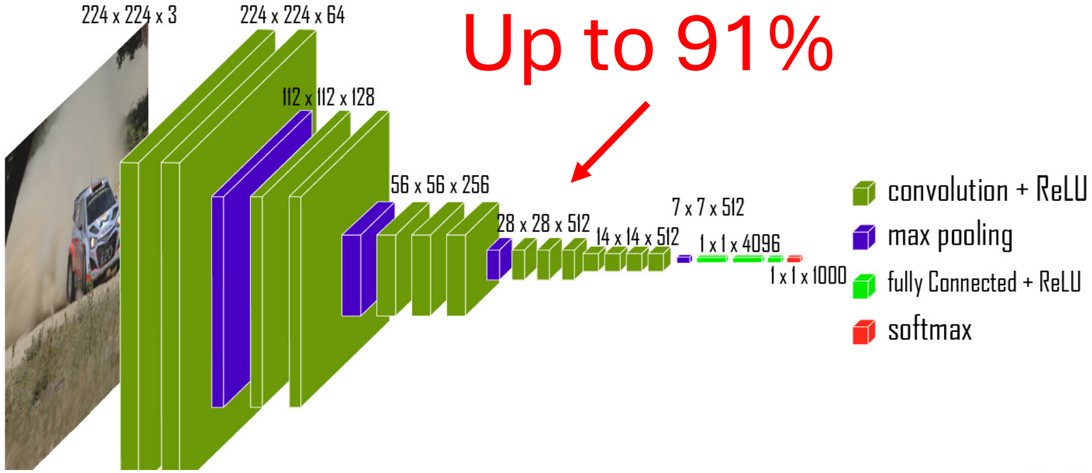
[2]University of Science and Technology of China, [3]MIT, [4]Peking University, [5]Meta

Communication Matters in DNN Training

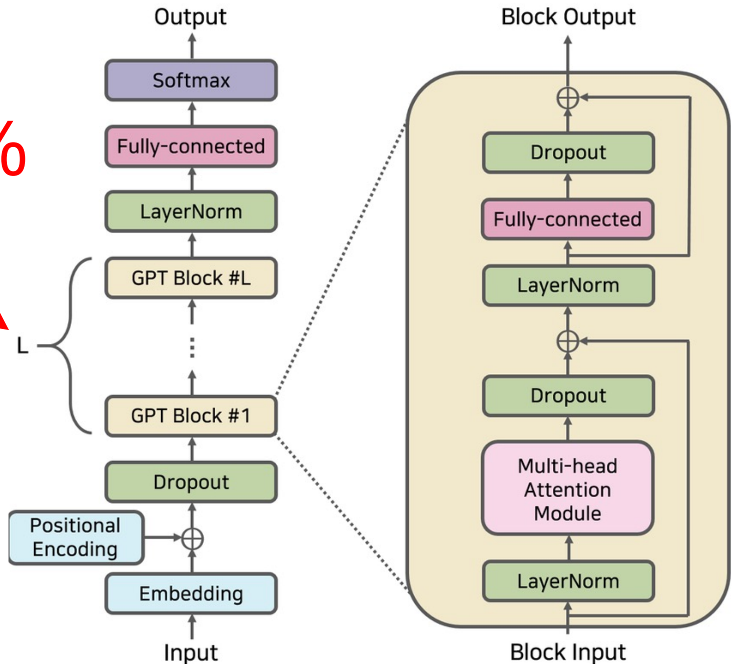
- Deep Neural Networks (DNNs) are increasingly adopted as fundamental building blocks in various modern services
- DNN training is essential in producing high-quality deep learning services
- Communication plays a significant role in distributed training

Up to 91%

Up to 49%



a VGG 16 model



a 1.5B GPT model

Prior Analysis Overlooks Critical Factors

- Many works aim to reduce the communication time in training, under specific model architectures or hardware platforms; do not provide a comprehensive overview of the communication characteristics
- Prior characteristics analysis works overlooks several critical factors

Cluster-level measurement works

- Viewe the entire training job as the basic unit
- Primarily assess cluster-level metrics like job completion time and cluster utilization;
- **Miss the fine-grained features within individual training jobs**

Works focusing on within-job scenarios

- **Also miss various key factors**
- Some only focus on data parallelism, ignoring model parallelism
- Some directly integrate the peak link capacity into analysis, overlooking the impact of various factors on bandwidth utilization.

Our Goal

We aim to conduct **a systematical exploration of the communication characteristics** of distributed training

- Our focus: individual job scenarios and fine-grained within-job features
- We analyze the communication through two aspects: (1) pattern and (2) overhead.

Pattern

High-level traffic attributes, such as predictability

Overhead

Metrics of communication time and communication ratio

Communication Pattern of Densely-activated Models

iSingLab HKUST

- Two primary elements of pattern: communication matrix and traffic volume

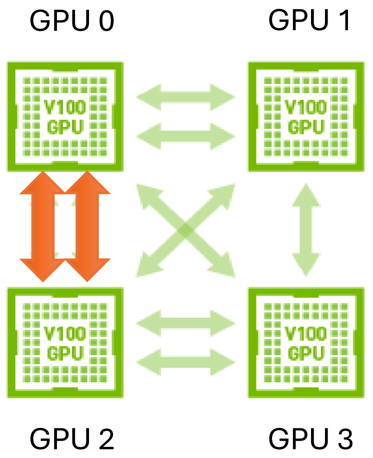
GPU Index

GPU Index	GPU 0	GPU 1	GPU 2	GPU 3
GPU 0	1	1	1	0
GPU 1	0	0		0
GPU 2	1	1	0	0
GPU 3	0	0	0	1

+ Traffic Volume

GPU Index

GPU Index	GPU 0	GPU 1	GPU 2	GPU 3
GPU 0	33	0	100	0
GPU 1	0	0	0	0
GPU 2	10	0.1	0	0
GPU 3	0	0	0	10



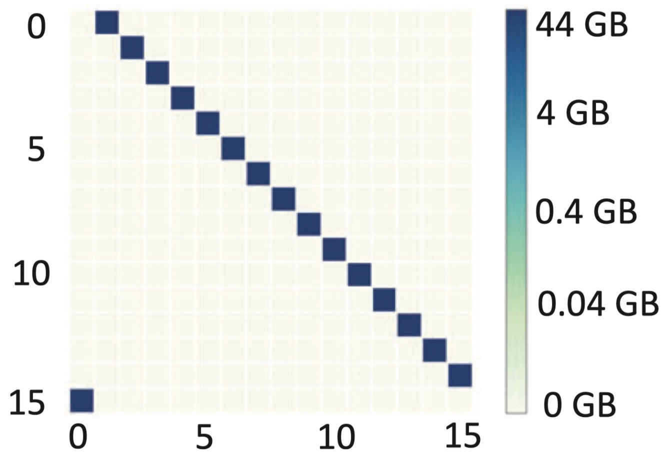
GPU 0 transmits an amount of traffic to GPU 2.

The traffic volume from GPU 0 to GPU 2 is 100.

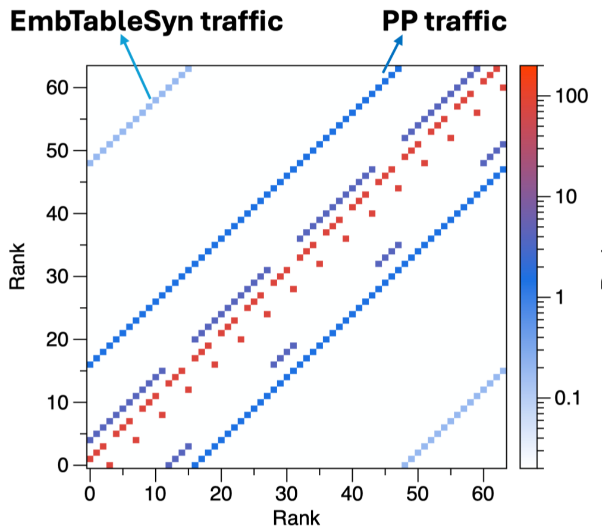
- For densely-activated models, both the communication matrix and traffic volume are predictable.

Communication Matrix

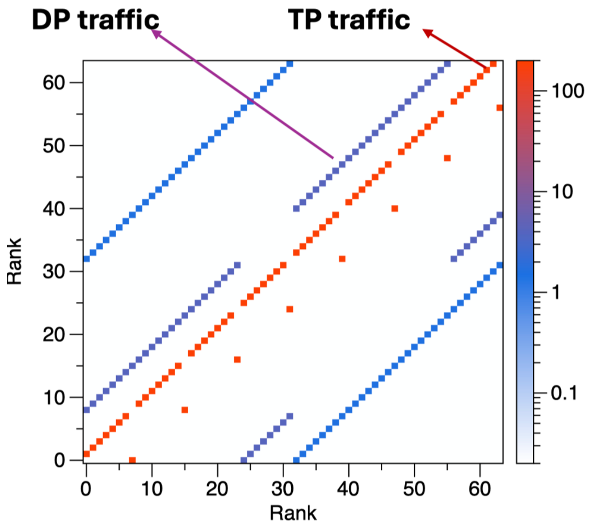
- Parallelism configuration determines the communication matrix.



Data parallelism
cited from TopoOpt[1]



PTD parallelism
(p, t, d) = (4, 4, 4)



PTD parallelism
(p, t, d) = (2, 8, 4)

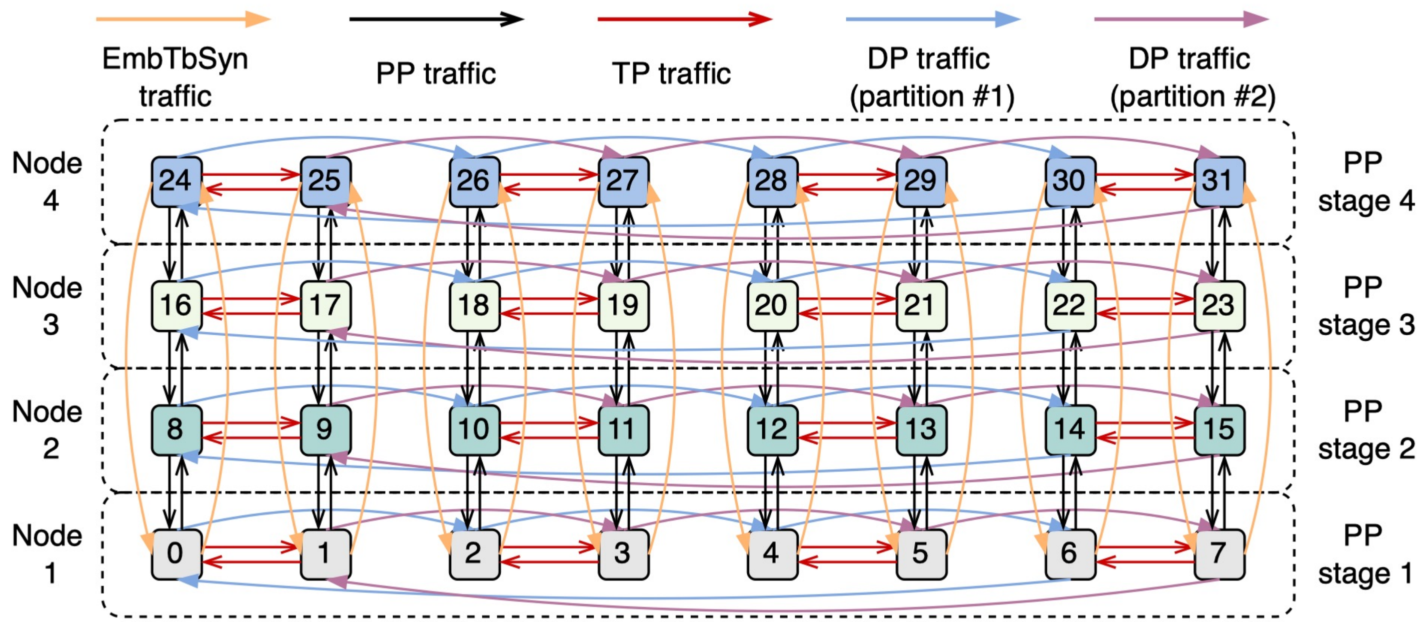
Purely DP models forms a diagonal line, regardless of the specific model architectures.

TP and DP traffic follow an AllReduce pattern structure. PP introduces P2P traffic between adjacent stages. EmbTableSyn traffic: aggregating the embedding gradients between the first and last PP stages

[1]TOPOOPT: Co-optimizing Network Topology and Parallelization Strategy for Distributed Training Jobs, NSDI 2023

Communication Matrix (Cont.)

- Given a parallelism configuration, communication matrix can be directly determined without running the model and conducting online profiling.
  - (1) Logical parallelism configuration
  - (2) Mapping principle from logical parallelism to physical hardware platform



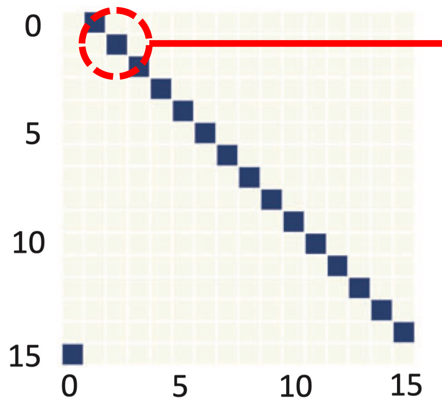
The GPU organization of a logical parallelism (p, t, d) = (4, 2, 4) on 32 GPUs

- GPUs → several **PP Stages**
- Within each stage → several **TP groups** and **DP groups**
- Each GPU simultaneously belongs to only one TP stage, one TP group, and one DP group

- Adjacent PP stages: **P2P traffic**
- TP groups: **AllReduce traffic**
- DP groups: **AllReduce traffic**

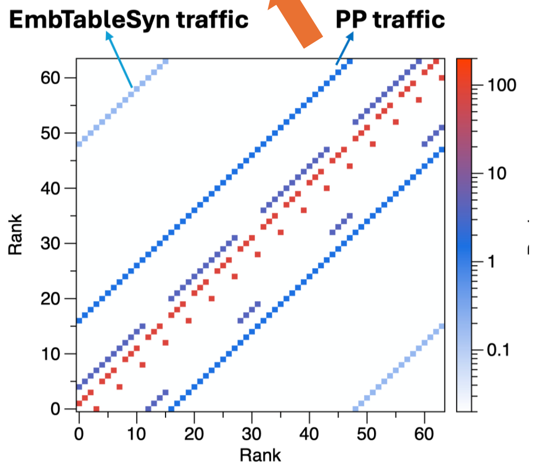
Traffic Volume

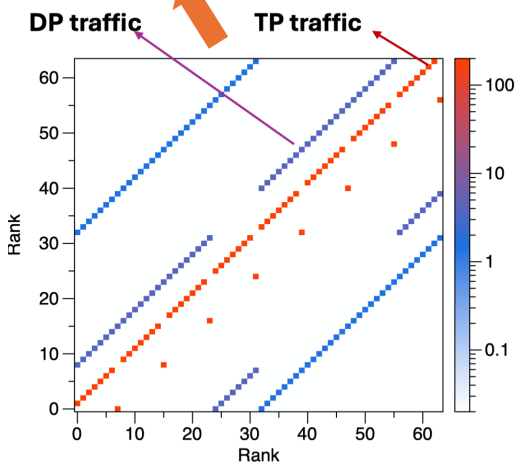
- Model's internal architecture influences the traffic volume on GPU pairs.
- Given the model architecture and parallelism configuration, the traffic volume is computable.



Purely DP models	Traffic volume = No. of parameters x parameter precision x $\frac{2(d-1)}{d}$
GPT models with PTD parallelism	Given the determined model architecture (N, l, h, s, gb, b, m) and parallelism configuration (p, t, d), the traffic volume on each GPU pair can be precisely calculated.

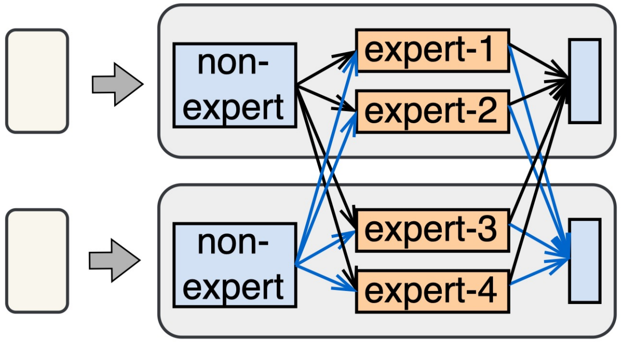
Notation	Explanation
p, t, d	(p, t, d) for the pipeline parallel size, tensor parallel size, and data parallel size, respectively.
N	Total number of model parameters.
l	Number of transformer block layers
h	Hidden size
s	Sequence length
gb, b	Global and micro-batch size, respectively
m	Number of micro-batches per iteration



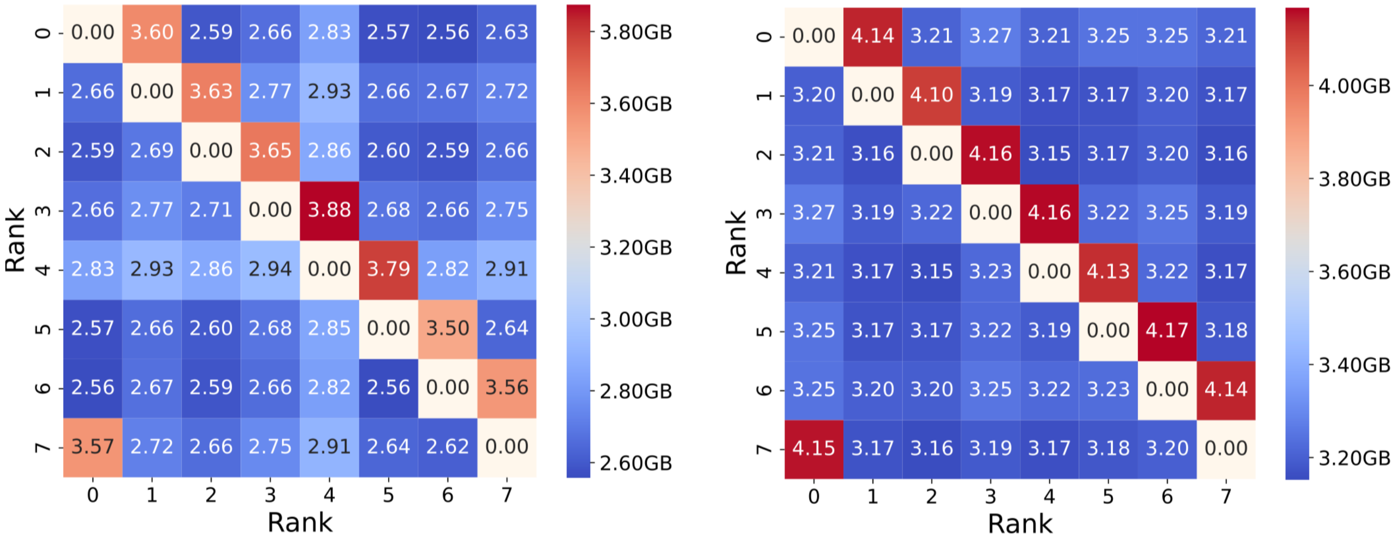


Communication Pattern of Sparsely-activated Models

- The MoE structure is a popular way to implement sparsely-activated models.
- Training large MoE models → expert parallelism (EP) → introducing AllToAll communication



- AllToAll traffic makes MoE training with dynamic communication patterns



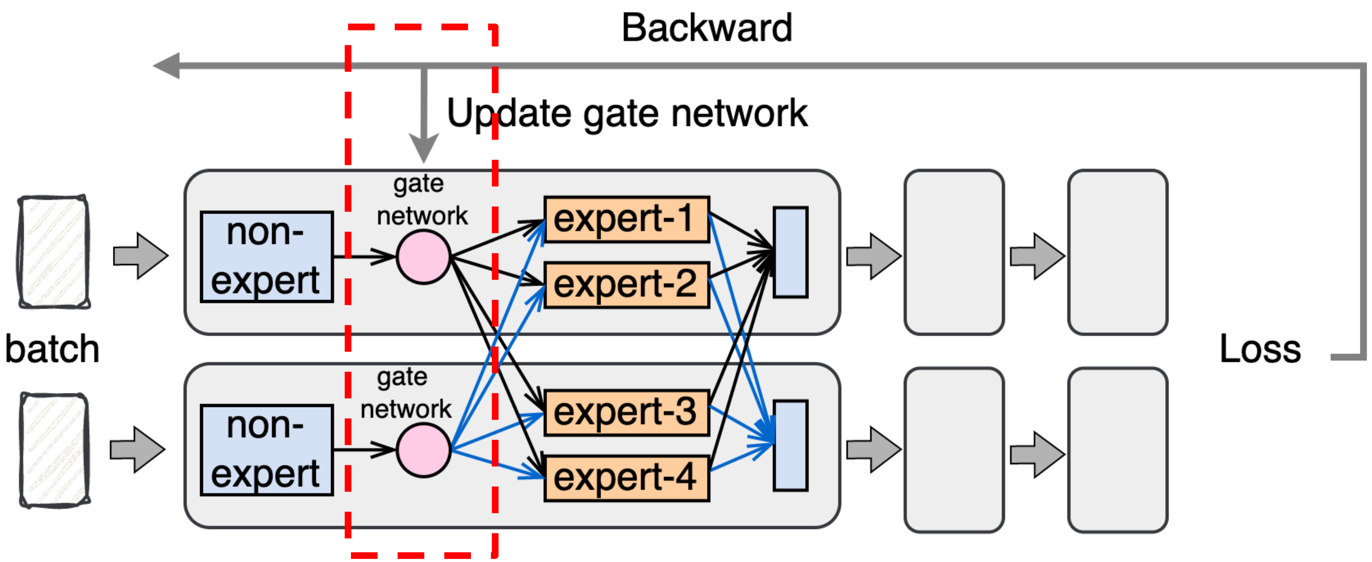
Traffic heatmaps of a 760M MoE model at different iterations

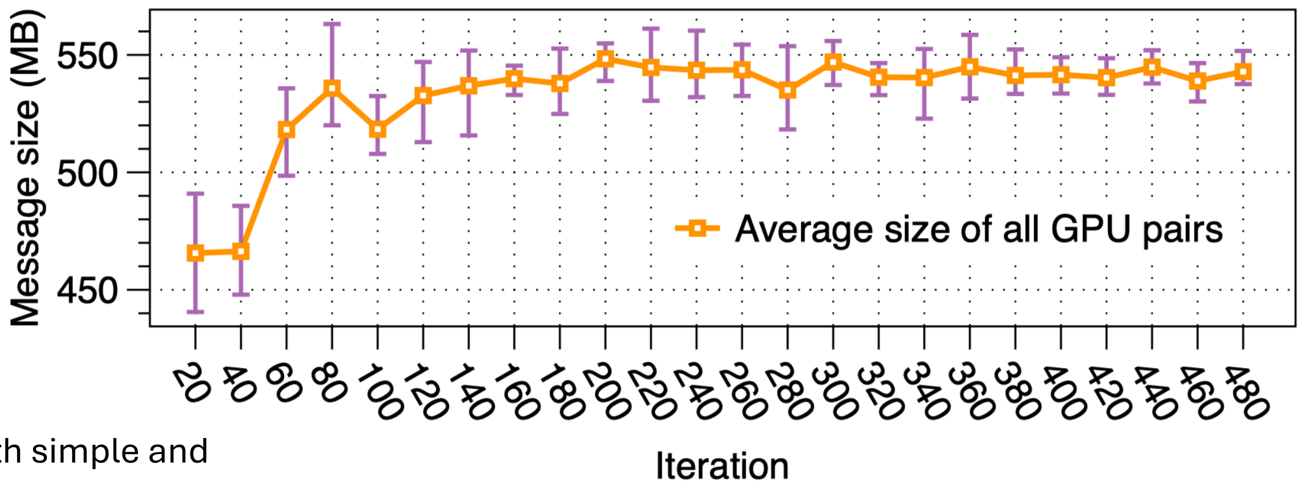
Red diagonal squares represent a combination of AllReduce traffic (DP) and AllToAll traffic (EP)

Blue squares indicate exclusively AllToAll traffic from EP

Semi-predictability of MoE Models

- The gate network is trained to achieve load balancing of traffic across experts[1].
  - The loss function is related to load balancing
- This leads to the increasing uniformity in AllToAll traffic patterns as training progresses.
- Average AllToAll traffic volume and variance during a MoE-1.3B model's training with $(e,\ d) = (8,8)$ [first 500 iterations]



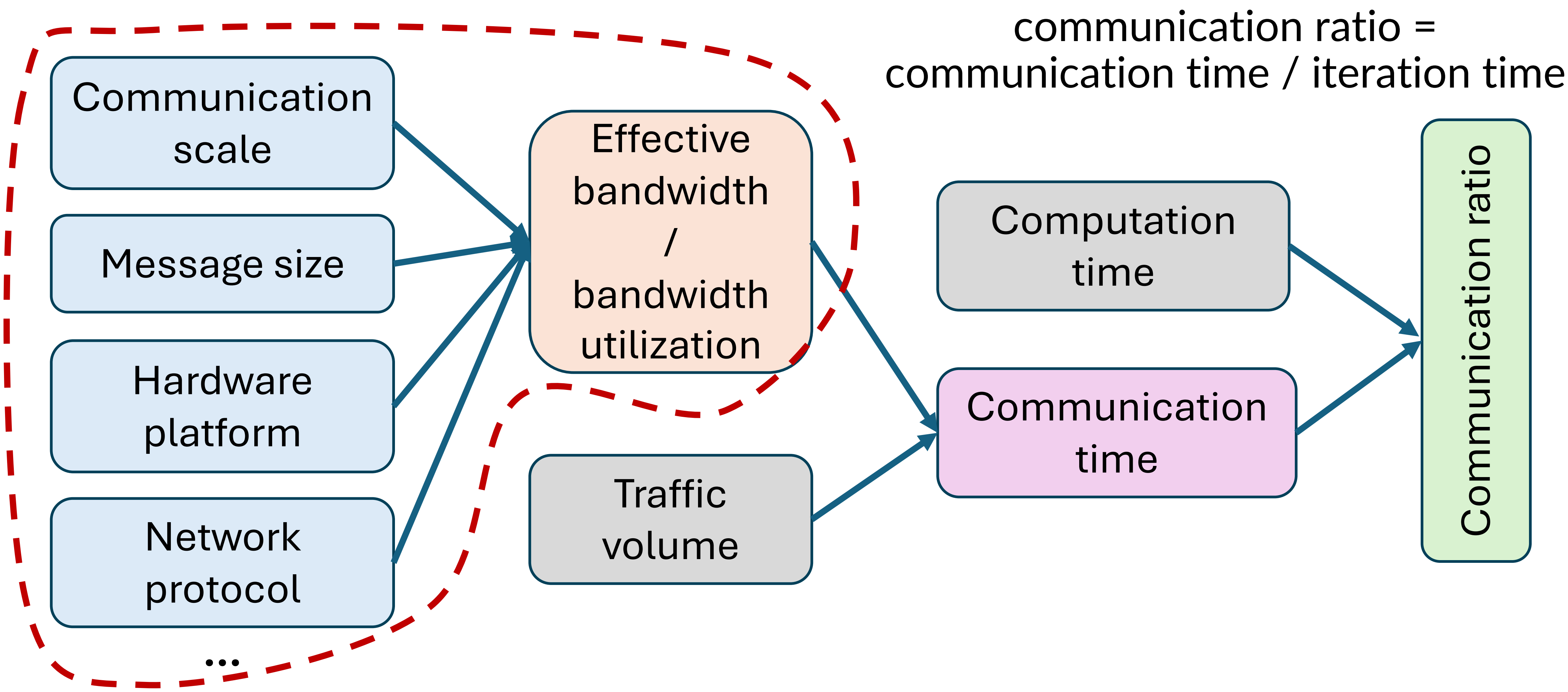


[1]Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research 2022

Other Characteristics

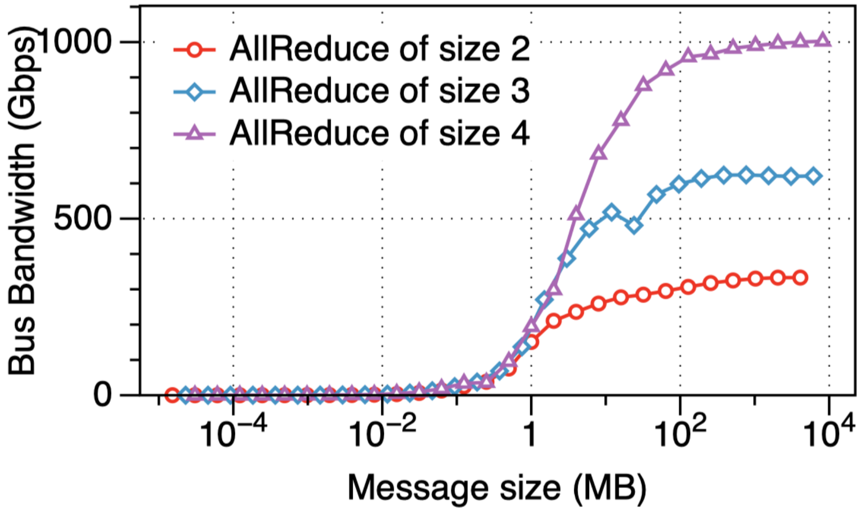
- Regularity (on-off pattern)
    - CASSINI: Network-Aware Job Scheduling in Machine Learning Clusters, NSDI 2024
    - …
- Low entropy
    - RDMA over Ethernet for Distributed AI Training at Meta Scale, SIGCOMM 2024
    - …
- Loss tolerance
    - Towards Domain-Specific Network Transport for Distributed DNN Training, NSDI 2024
    - …
- …

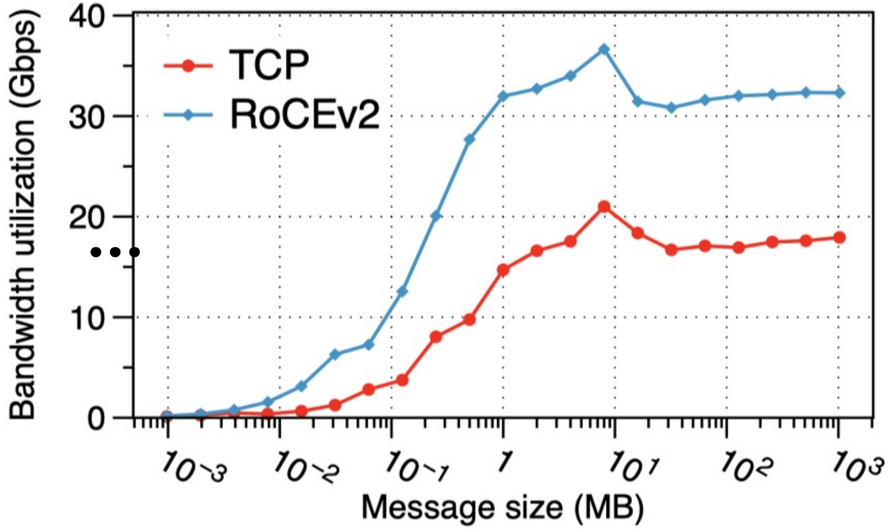
Factors on Communication Overhead

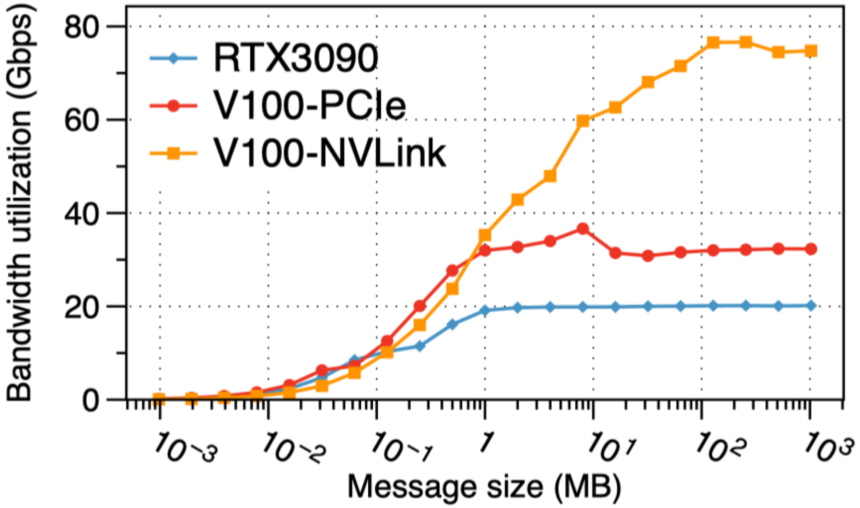


Factors on Effective Bandwidth

- Communication scale
- Message size
- Hardware platform
- Network protocol



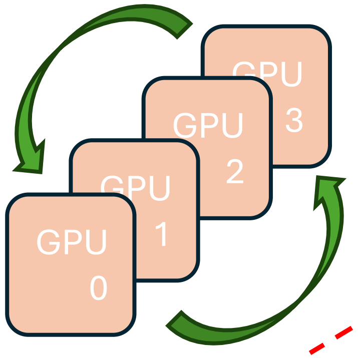




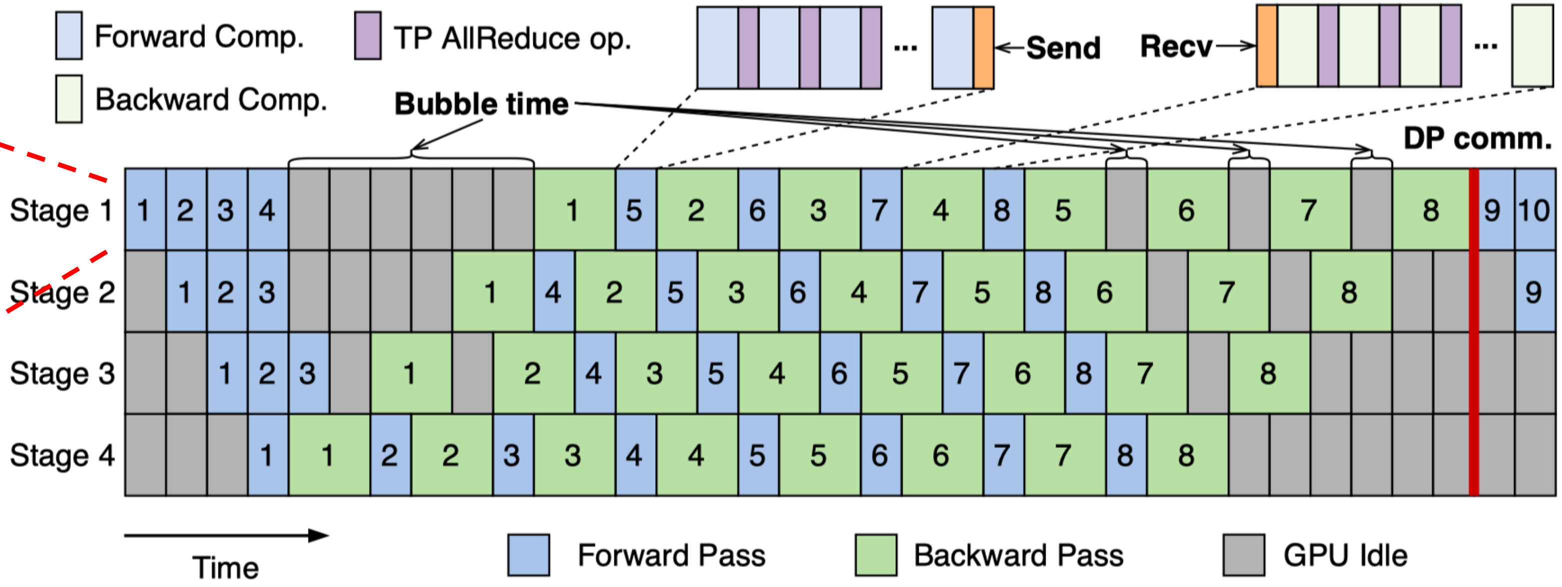
Communication Overhead Estimation

- We propose an analytical formulation to estimate the communication overhead (time & ratio) of GPT models with PTP parallelism

TP AllReduce



a TP group at stage 1



4 PP stages, 8 micro-batches

- Iteration time = total working time of any GPUs at stage 1 (e.g., GPU 0)

$$T_{iter} = T_{comp} + T_{TP} + T_{PP} + T_{DP} + T_{bubble}$$

Communication Overhead Estimation (Cont.)

iSingLab HKUST

- T_{comp}

μ: GPU utilization rate

Computation requirement for each micro-batch[1]:

$$FLOP^{mb}_{required} = 8 \times \frac{N}{p \times t} \times b \times s$$

Time:

$$T_{comp} = \frac{m \times FLOP^{mb}_{required}}{\mu F} = \frac{8m \times N \times b \times s}{p \times t \times \mu F}$$

- T_{bubble}

$$T_{bubble} = (p-1) \times (T^{mb}_{comp} + T^{mb}_{PP} + T^{mb}_{TP})$$

$$R_{bubble} = T_{bubble} / T_{iter} \approx (p-1)/(p-1+m)$$

[1]Efficient Large-scale Language Model Training on GPU Clusters Using Megatron-LM, SC 2021

- T_{TP}, T_{PP}, T_{DP}

$$T_{DP} = \frac{2N}{p \times t} \times \frac{2(d-1)}{d \times C_{DP}}$$

Traffic volume

$$T_{PP} = m \times T^{mb}_{PP} = m \times \frac{2 \times 2bsh}{C_{PP}}$$

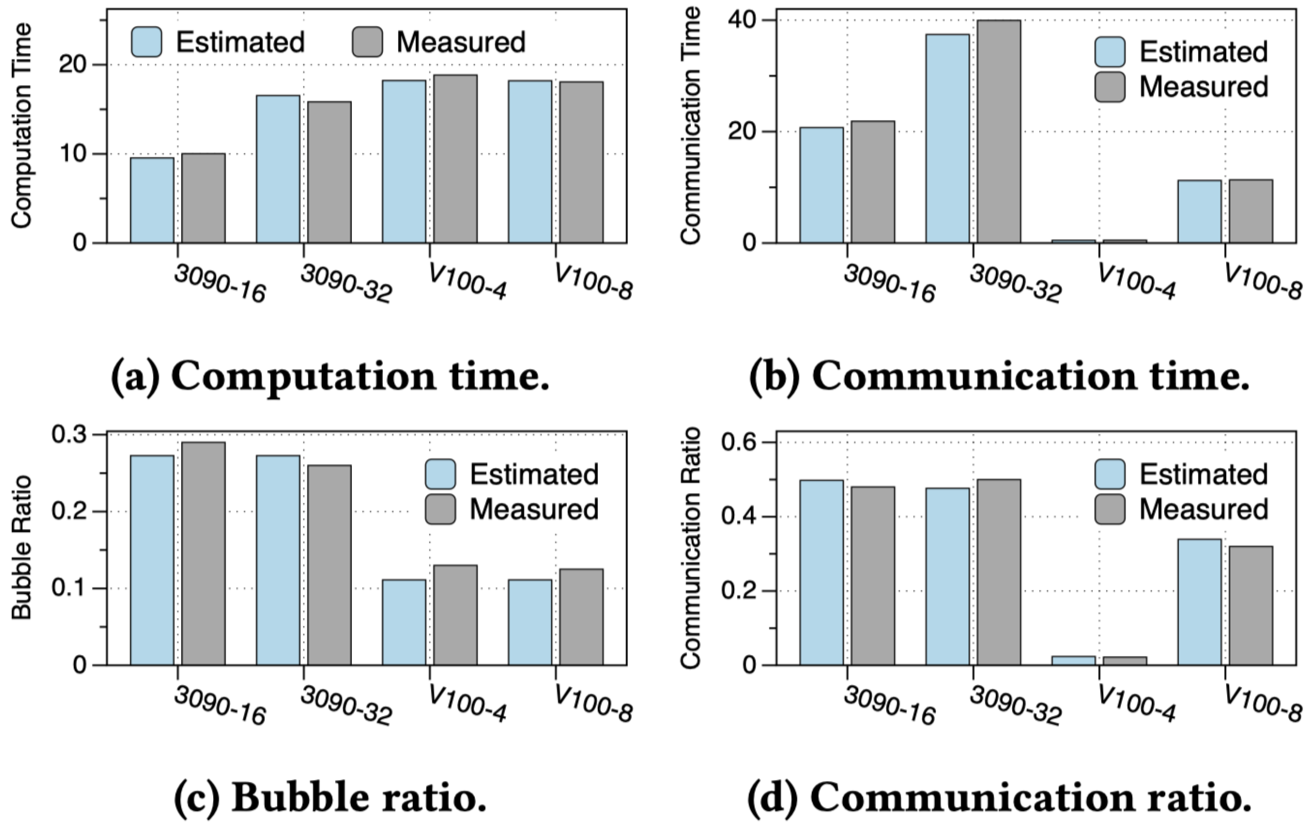
Effective bandwidth

$$T_{TP} = m \times T^{mb}_{TP} = m \times \frac{l}{p} \times \frac{6 \times 2bsh \times 2(t-1)}{t \times C_{TP}}$$

Notation	Explanation
p, t, d	(p, t, d) for the pipeline parallel size, tensor parallel size, and data parallel size, respectively.
N	Total number of model parameters.
l	Number of transformer block layers
h	Hidden size
s	Sequence length
gb, b	Global and micro-batch size, respectively
m	Number of micro-batches per iteration
$C_{TP,PP,DP}$	Effective bandwidth utilization of TP, PP, DP
F	GPU computation capacity (*i.e.*, peak FP16 FLOP/s)

Accuracy of Estimation

- Estimations from analytical formulation vs. measured realistic data
- Separately evaluate T_{comp}, T_{comm} (T_{TP}+ T_{PP}+ T_{DP}), $R_{comm}(\frac{T_{comm}}{T_{iter}})$, and R_{bubble}
- Four experimental configurations:
  - 16 RTX3090s, 1.5B GPT model
  - 32 RTX3090s, 3B GPT model
  - 4 V100s, 1.5B GPT model
  - 8 V100s, 3B GPT model
- Config. of μ and C (C_{TP}, C_{DP}, C_{PP}): apply a μ of 0.3 for RTX3090 and 0.4 for V100; C is profiled using NCCL micro-benchmarks
- The formulation achieves ~90% accuracy across our experiments.



(a) Computation time. (b) Communication time. (c) Bubble ratio. (d) Communication ratio.

Conclusion

- We present a comprehensive analysis of the traffic predictability of densely-activated models and show the existence of dynamic traffic pattern and increasing uniformity in MoE model training.
- We experimentally evaluate the influence of various factors on the effective bandwidth (further influencing the communication overhead).
- We propose an analytical formulation to estimate communication overhead for GPT models

A systematical exploration is **still ongoing**, and the results and analysis presented in the APNET paper are quite preliminary.

Future Works

- (1) Broaden our experimental setting to incorporate more advanced GPUs and larger training scales to verify our current findings
- (2) Conduct an in-depth exploration on two critical factors used in our analytical formulation
  - GPU utilization rate (μ)
  - Effective bandwidth (C)
- Maybe more ...

Thank you.