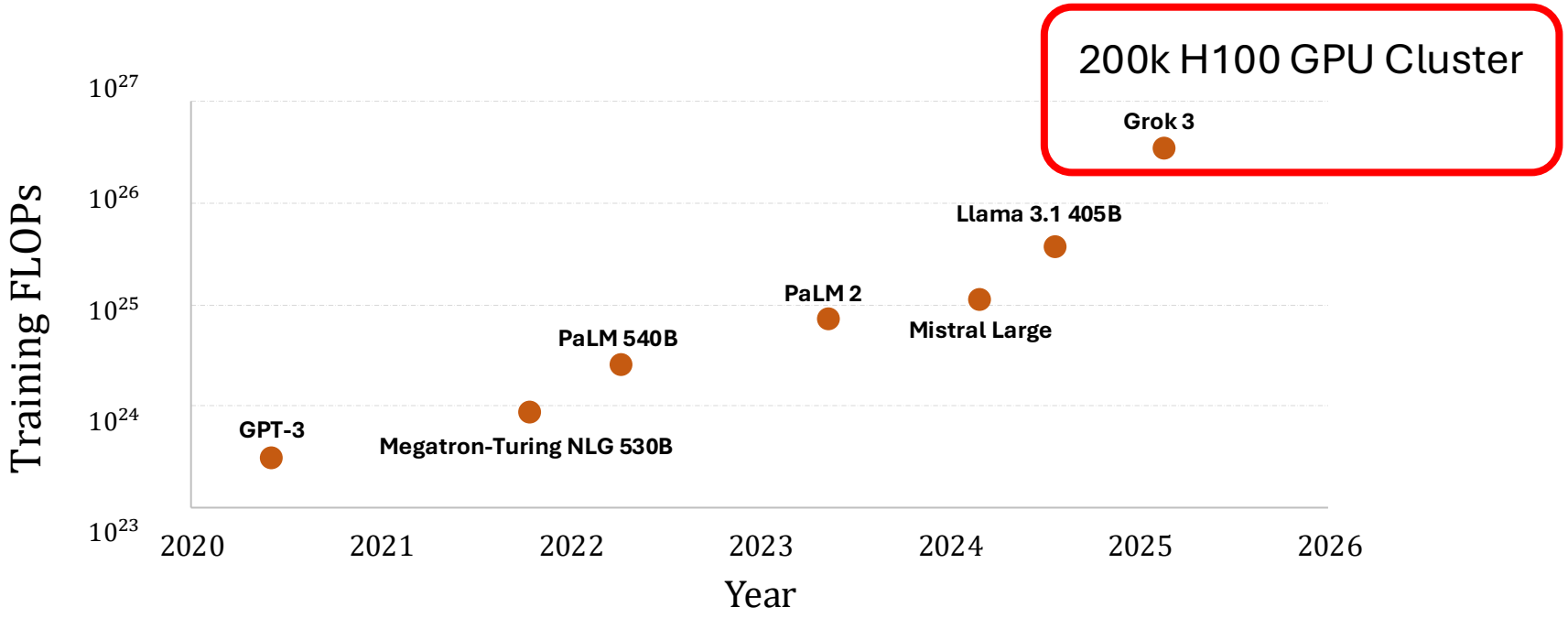
Aggregate Local, Sync Global: A Hierarchical Approach to Efficient Geo-Distributed LLM Training

Francesco De Luca [*¶], **Francesco De Nadai** [*¶], Mariano Scazzariello [‡],
Tommaso Caiazzi [¶], Alireza Farshin [†], Marco Chiesa [§], Giuseppe Di Battista [¶]

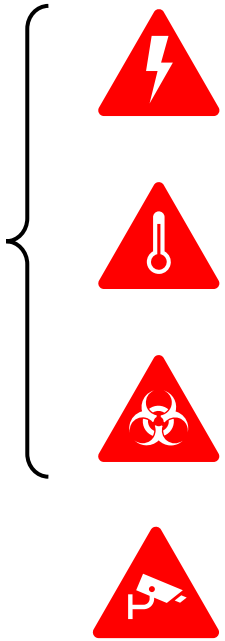
[¶] Roma Tre University - [‡] RISE Research Institutes of Sweden - [†] NVIDIA - [§] KTH Royal Institute of Technology

NetCompute 2026 | Tokyo, Japan | May 18, 2026

Training Frontier LLMs Demand 1000× More Compute Than 2020



Single-Datacenter Training is Infeasible



Power provisioning

Thermal dissipation

Environmental sustainability

Data sovereignty

Single-Datacenter Training is Infeasible

Power provisioning

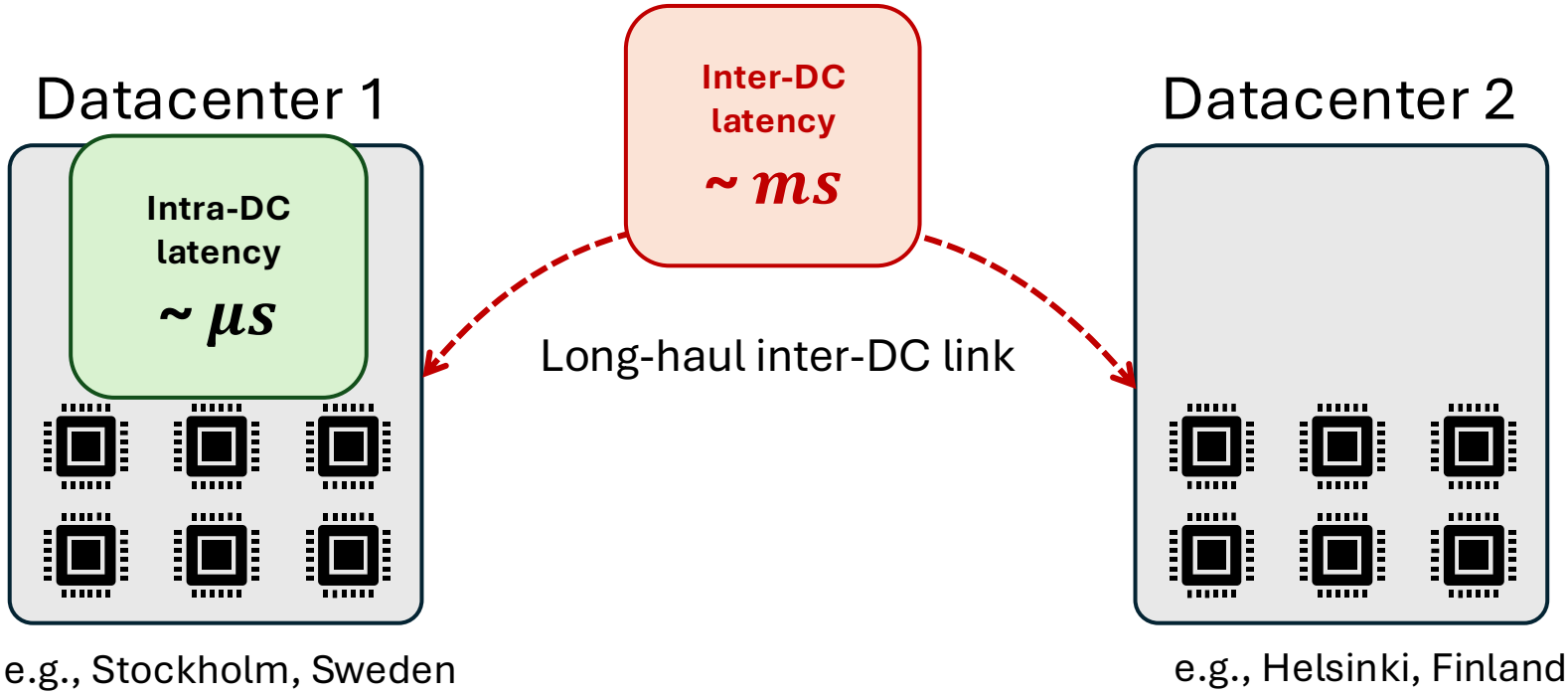
Thermal dissipation

Environmental sustainability

Data sovereignty

Multi-Datacenter training!

Geo-Distributed Training Introduces a New Bottleneck



Inter-DC communication becomes a dominant factor!

Existing Solutions Fall Short in Multi-DC Environments

Current Collective Communication Libraries (CCLs), e.g., TACCL [1] (Topology-Aware CCL) and TE-CCL [2] (Traffic Engineering CCL), are designed and optimized exclusively for single-site deployments, often hitting hard scalability limits

[1] **TACCL**: Shah et al., *"TACCL: Guiding Collective Algorithm Synthesis using Communication Sketches"*, NSDI 2023

[2] **TE-CCL**: Liu et al., *"Rethinking Machine Learning Collective Communication as a Multi-Commodity Flow Problem"*, ACM SIGCOMM 2024

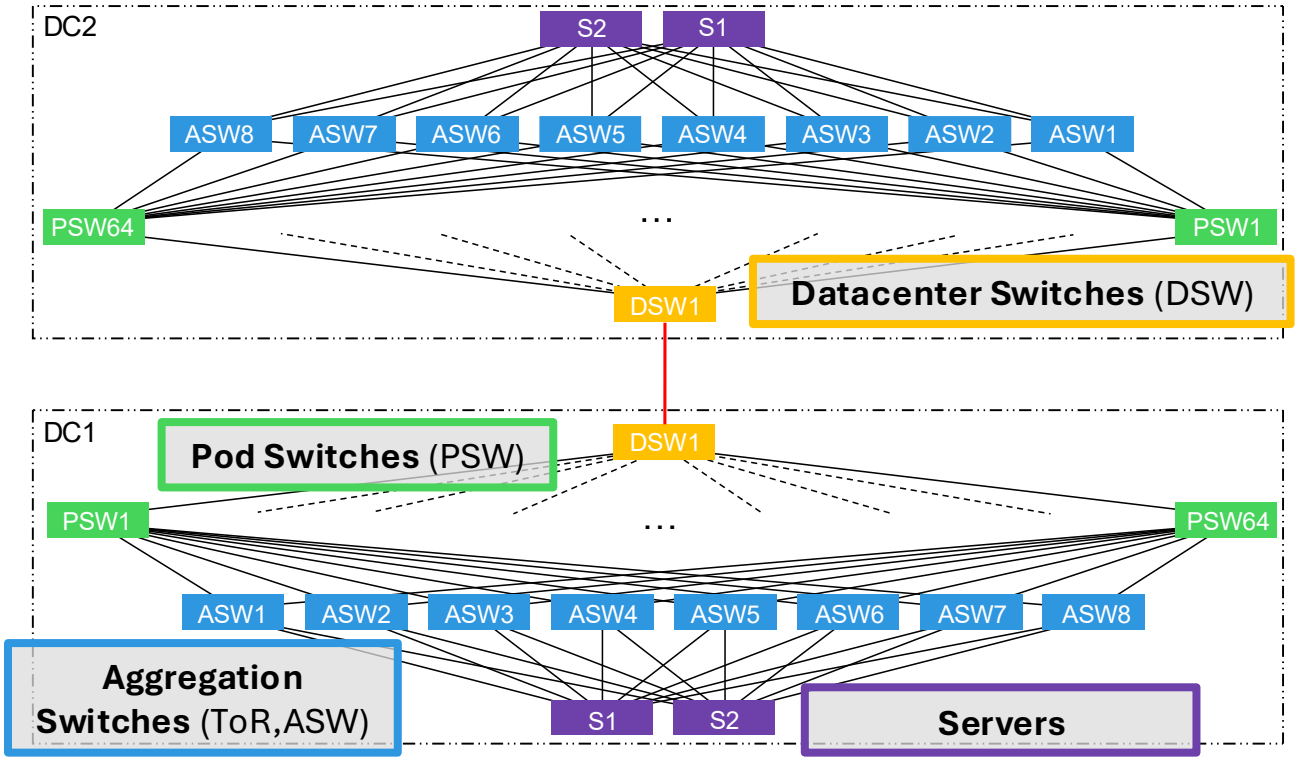
Existing Solutions Fall Short in Multi-DC Environments

Current Collective Communication Libraries (CCLs), e.g., TACCL [1] (Topology-Aware CCL) and TE-CCL [2] (Traffic Engineering CCL), are designed and optimized exclusively for single-DC deployments, often hitting hard scalability limits

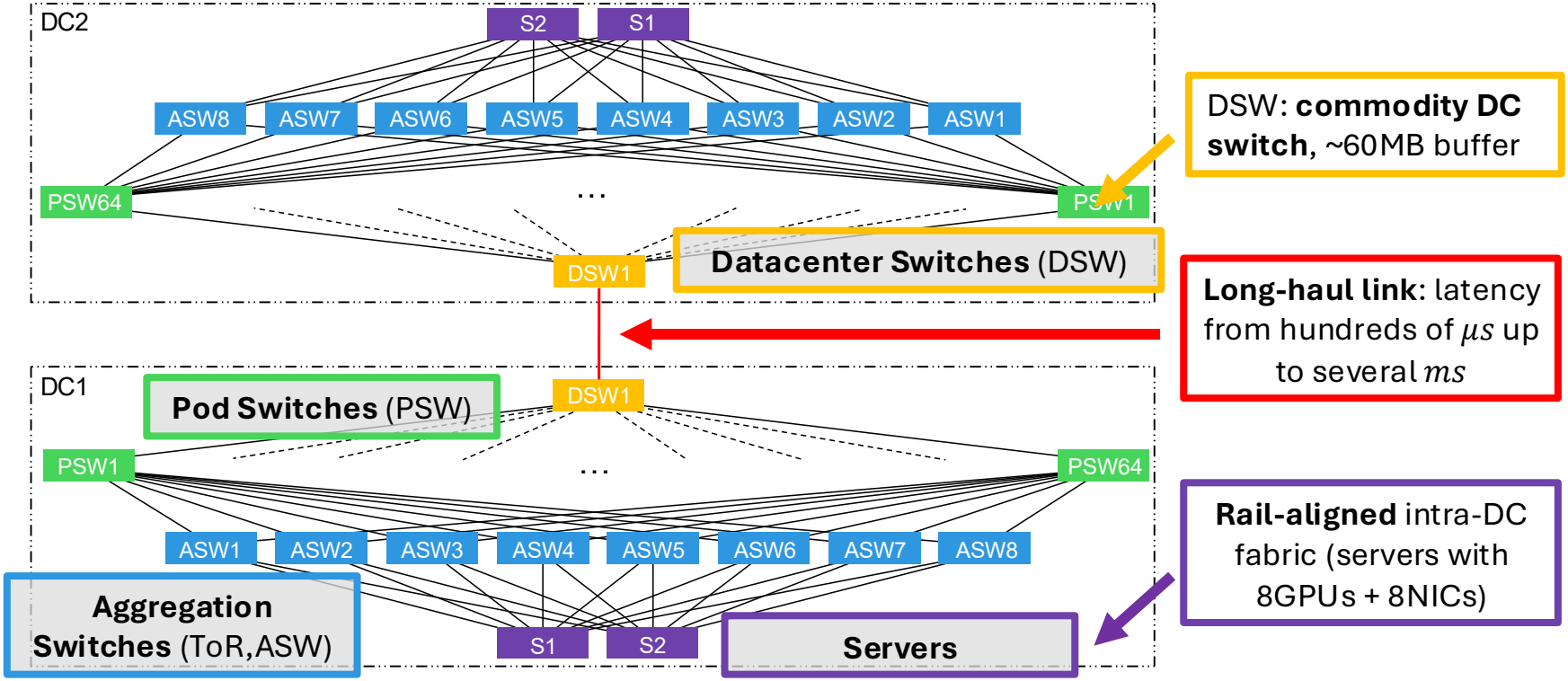
High-level training frameworks, e.g., Megatron-LM [3] or NVIDIA Nemo [4], while technically supporting multi-DC training, leave key network optimizations unimplemented in their standard communication backend

[3] **Megatron-LM**: M. Shoeybi et al., "*Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*", arXiv preprint, arXiv:1909.08053, 2019
[4] **NVIDIA Nemo**: O. Kuchaiev et al., "*NeMo: a toolkit for conversational AI and large language models*", arXiv preprint arXiv:1909.09577, 2019

Multi-Site Network Topology



Multi-Site Network Topology



Our Setup

Priority Flow Control [5] (PFC) needs buffer space for all **in-flight packets before a PAUSE frame takes effect**. On long-haul links, this flight time is too large and **many packets are already in flight**. DSW needs huge buffers to absorb them.

Unsustainable!

[5] **Priority Flow Control (PFC):** IEEE Standard 802.1Qbb, 2011. Data Center Bridging (DCB) Task Group.

Our Setup

- **Lossy RDMA**: packets are dropped, e.g., when the DSW buffer is full
- NICs recover via **Selective Repeat**

- DSW does not generate PFC frames
- **No risk of PFC storms** [6] propagating back into the intra-DC fabric

[6] **PFC Storms:** C. Guo et al., "RDMA over Commodity Ethernet at Scale", ACM SIGCOMM 2016.

Our Setup

Cross-DC Data Parallelism

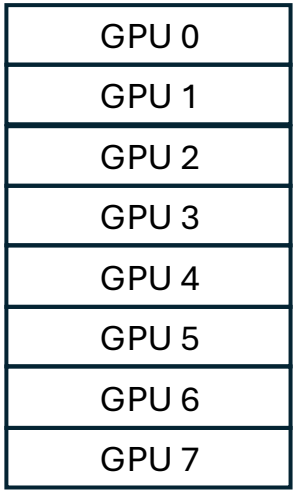
Model replicas are distributed across sites,
leading to **inter-DC gradient synchronization**

Our Setup

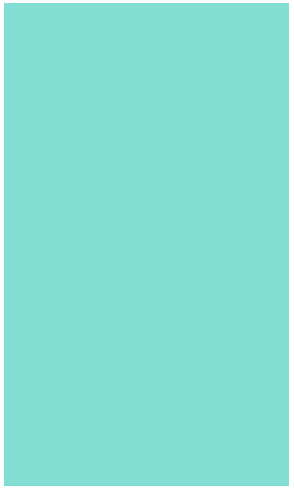
Reference Setup

$$\#GPU = 32$$
$$\#DC = 2$$

$$Tensor\_Parallelism = 8$$
$$Pipeline\_Parallelism = 1$$
$$Data\_Parallelism = 4$$



8 GPUs server



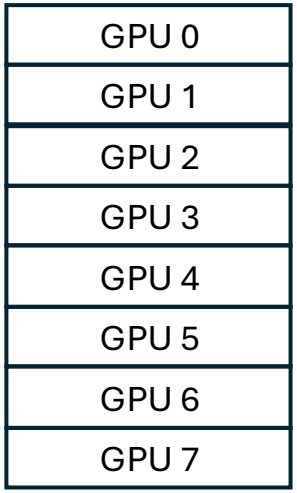
model weights

Our Setup

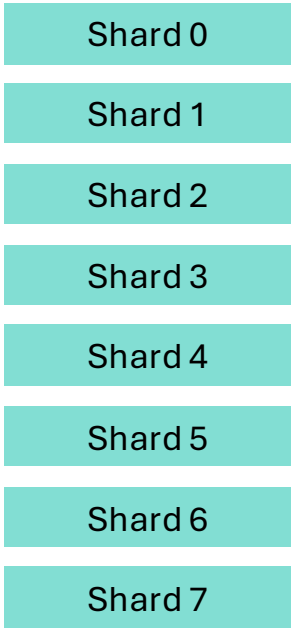
Reference Setup

$$\#GPU = 32$$
$$\#DC = 2$$

$$Tensor\_Parallelism = 8$$
$$Pipeline\_Parallelism = 1$$
$$Data\_Parallelism = 4$$

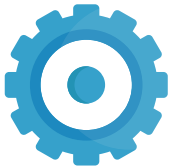


8 GPUs server



sharded model weights

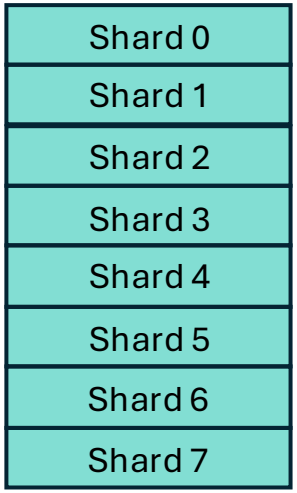
Our Setup



Reference Setup

$$\#GPU = 32$$
$$\#DC = 2$$

$$Tensor\_Parallelism = 8$$
$$Pipeline\_Parallelism = 1$$
$$Data\_Parallelism = 4$$



8 GPUs server

Each GPU hosts a single shard of the parameters!

Each servers hosts an entire replica of the model!

The Industry Standard: Megatron-LM

Communication is structured along two dimensions:

① Intra-Replica — Tensor Parallelism (TP)

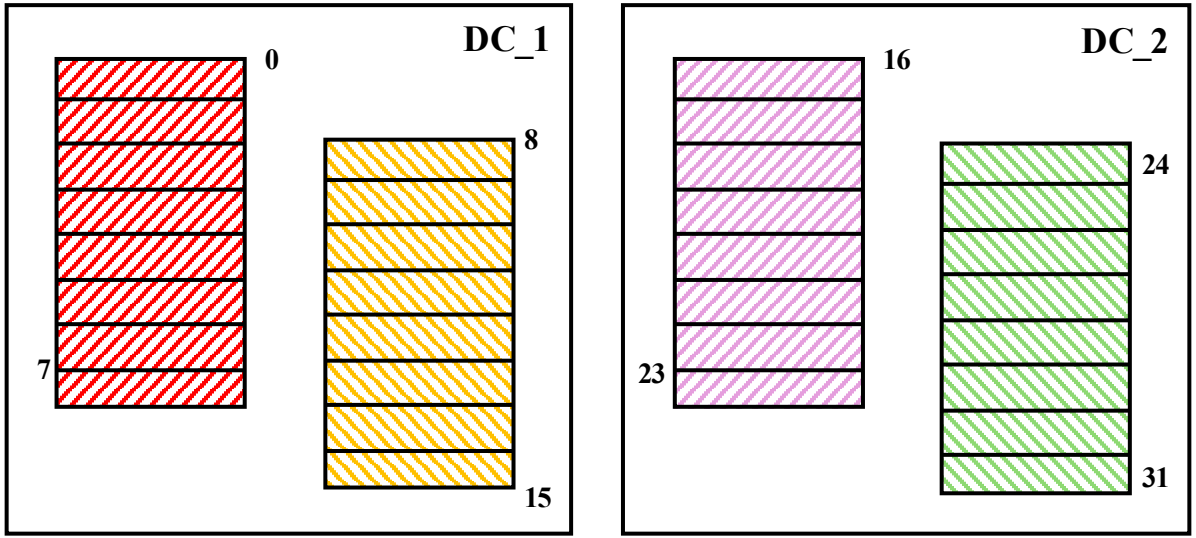
Groups of 8 contiguous GPUs share one model replica. Weights are sharded, corresponding GPUs across replicas hold the same weight shard. Communication stays intra-node (NVLink).

② Inter-Replica — Data Parallelism (DP)

Synchronizes gradients across replicas via All-Reduce. Each DP ring links corresponding GPUs of each replica, one ring per intra-replica rank. Spans inter-node and inter-DC links.

Baseline Multi-DC Approach

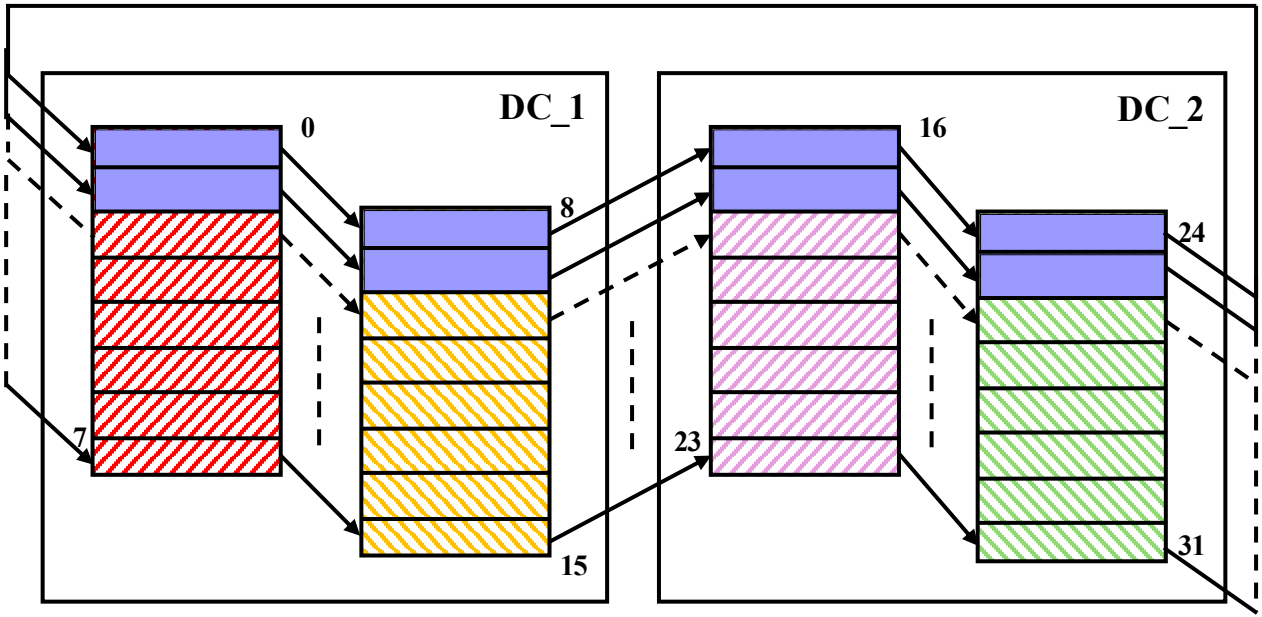
$\#GPU = 32$
$\#DC = 2$
$TP = 8$
$PP = 1$
$DP = 4$



① **Intra-Replica — Tensor Parallelism (TP)**

Baseline Multi-DC Approach

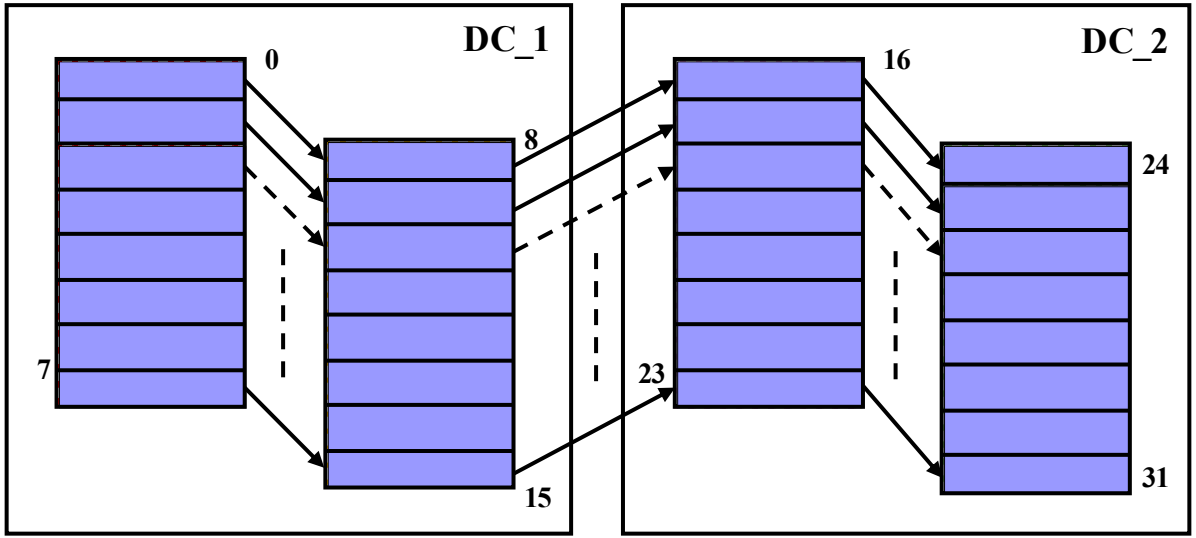
$$\#GPU = 32$$
$$\#DC = 2$$
$$TP = 8$$
$$PP = 1$$
$$DP = 4$$



2 **Inter-Replica — Data Parallelism (DP)**

Baseline Multi-DC Approach

$$\#GPU = 32$$
$$\#DC = 2$$
$$TP = 8$$
$$PP = 1$$
$$DP = 4$$



② **Inter-Replica — Data Parallelism (DP)**

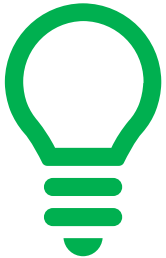
Baseline Multi-DC Approach

Site-local communications are affected by inter-DC link latency!

Ring’s dimension increases linearly with the number of replicas.

What happens if we decouple the DP communication in two distinct tiers: intra-DC and inter-DC?

Hierarchical Multi-DC Approach



We designate a **leader model replica** within each DC: its GPUs are **solely responsible for synchronizing gradients across sites**, while all non-leader replicas participate only in intra-DC communication.

Hierarchical Multi-DC Approach

The DP synchronization process consists of **three steps**:

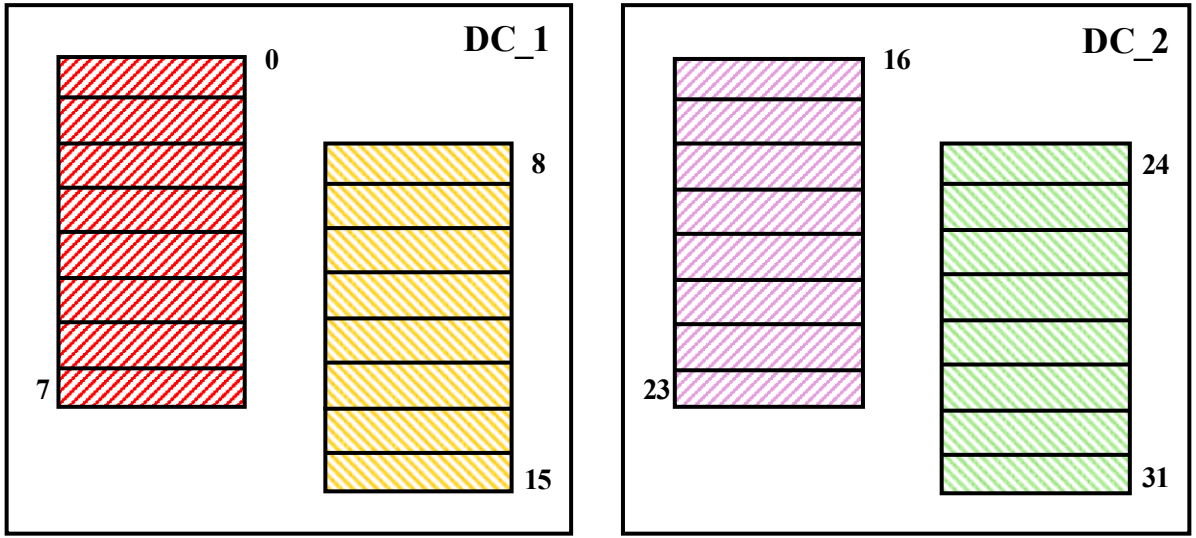
1. **Intra-DC Replicas Synchronization:** all model replicas locally synchronize their gradients

2. **Leader Replicas Inter-DC Synchronization:** to obtain a globally aggregated gradient

3. **Intra-DC Broadcast:** leader replica broadcasts the globally aggregated gradient to the non-leader replicas

Hierarchical Multi-DC Approach

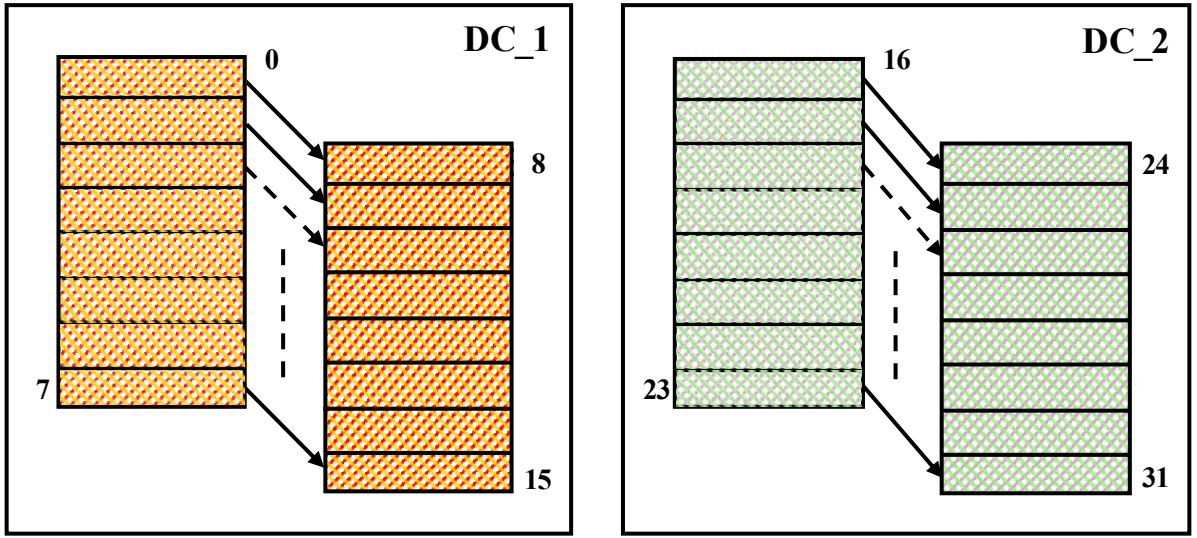
$$\#GPU = 32$$
$$\#DC = 2$$
$$TP = 8$$
$$PP = 1$$
$$DP = 4$$



① **Intra-DC Replicas Synchronization**

Hierarchical Multi-DC Approach

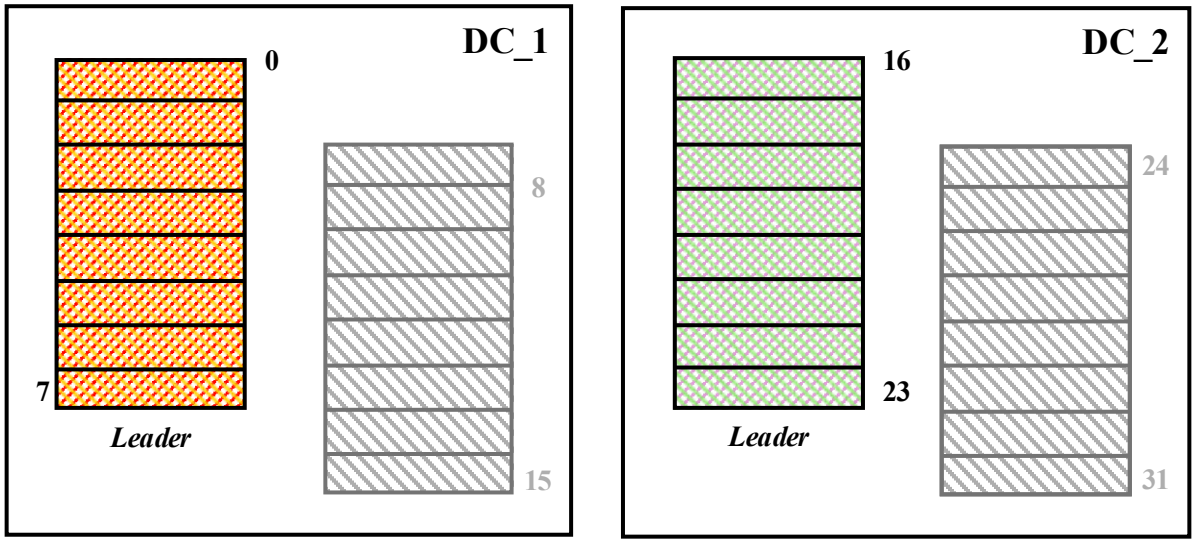
$\#GPU = 32$
$\#DC = 2$
$TP = 8$
$PP = 1$
$DP = 4$



① **Intra-DC Replicas Synchronization**

Hierarchical Multi-DC Approach

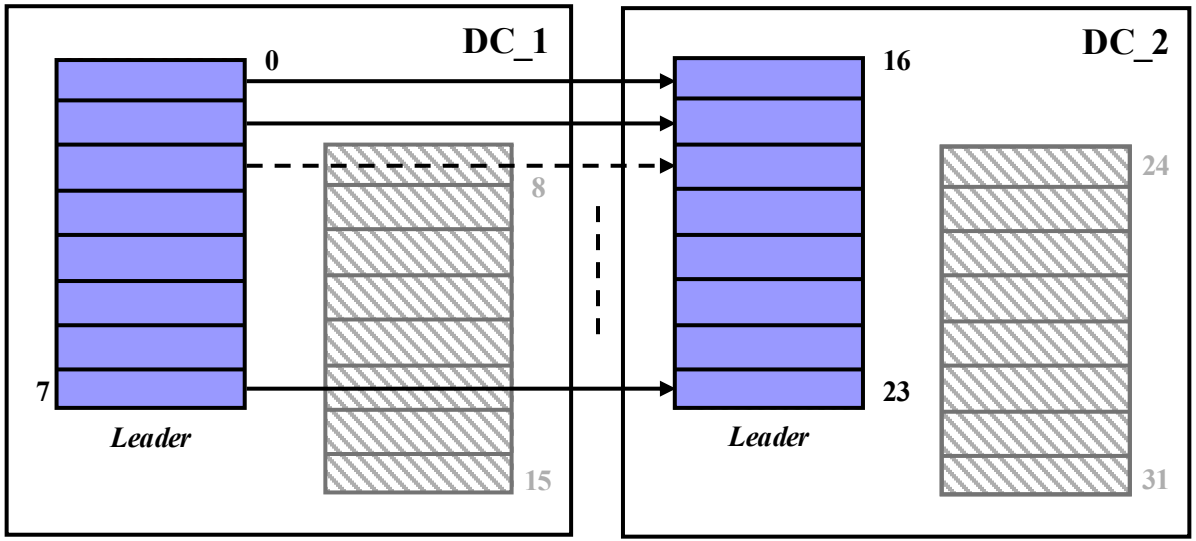
$\#GPU = 32$
$\#DC = 2$
$TP = 8$
$PP = 1$
$DP = 4$



2 **Leader Replicas Inter-DC Synchronization**

Hierarchical Multi-DC Approach

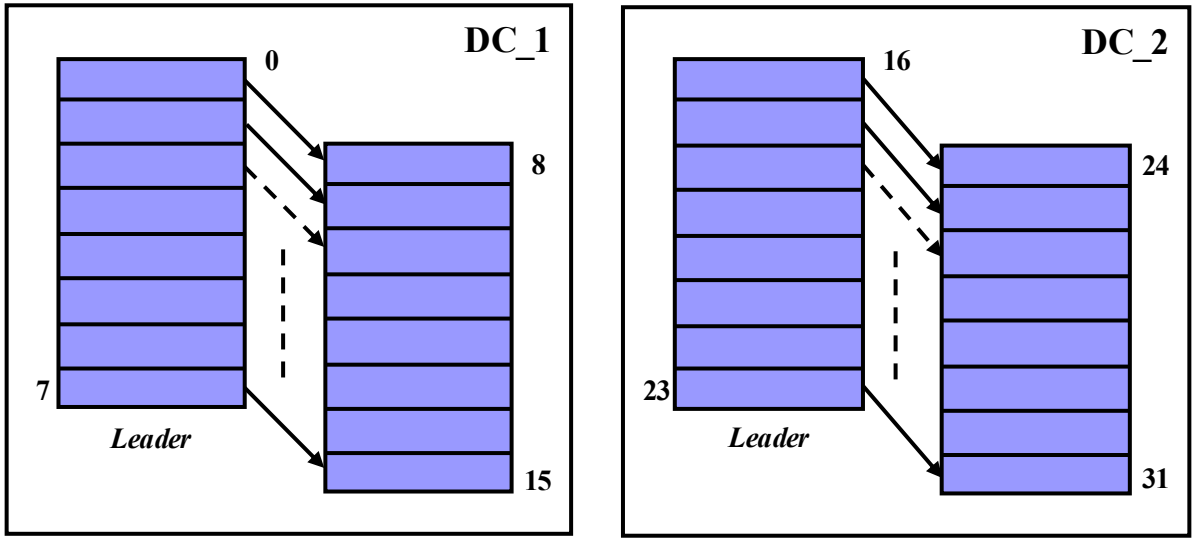
$$\#GPU = 32$$
$$\#DC = 2$$
$$TP = 8$$
$$PP = 1$$
$$DP = 4$$



② Leader Replicas Inter-DC Synchronization

Hierarchical Multi-DC Approach

$$\#GPU = 32$$
$$\#DC = 2$$
$$TP = 8$$
$$PP = 1$$
$$DP = 4$$



3 **Intra-DC Broadcast**

Experimental Setup: Simulation

SimAI [1] + Multi-DC support and Selective Repeat (lossy RDMA)

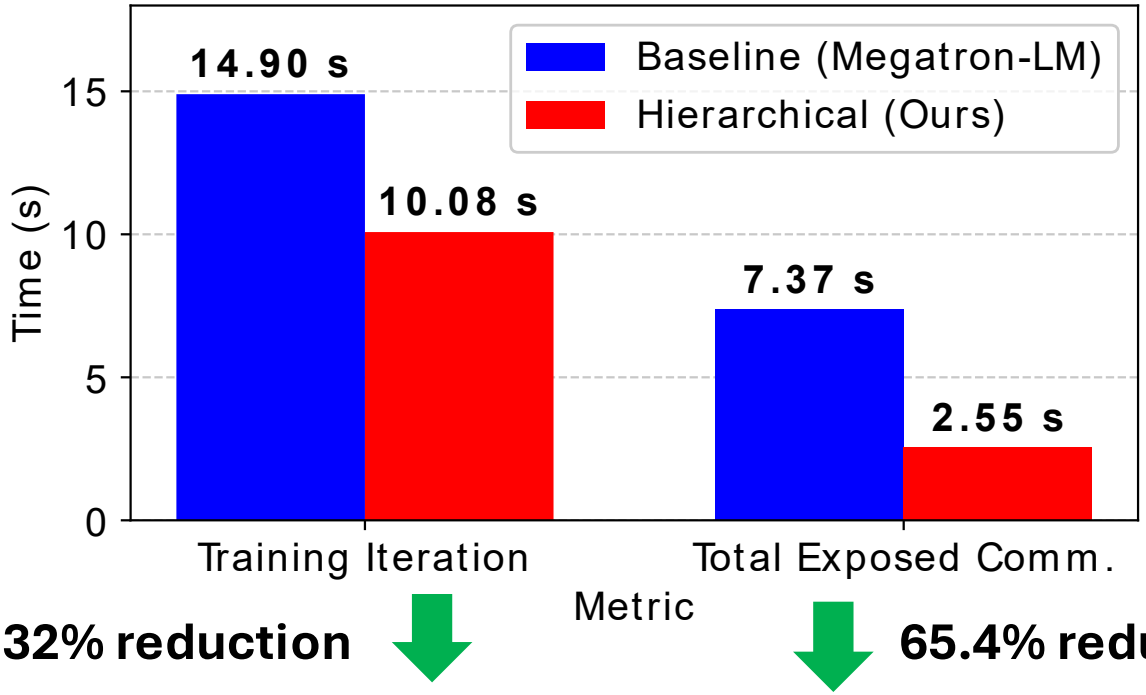
Evaluation Metrics → Training Iteration Time

Evaluation Metrics → Flow Completion Time

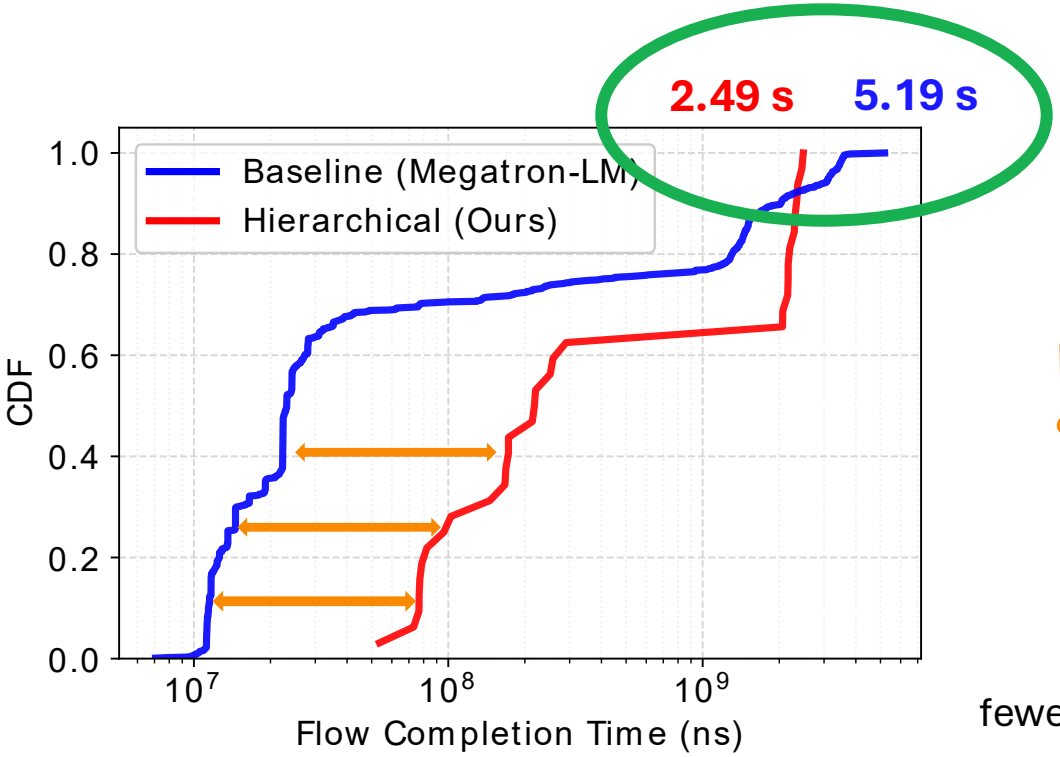
[1] **SimAI**: X. Wang et al., "*SimAI: Unifying Architecture Design and Performance Tuning for Large-Scale Large Language Model Training with Scalability and Precision*", NSDI 25

Evaluation Results

256 GPUs (128/DC) · 400Gbps inter-DC 500μs · 400Gbps intra-DC 1μs



Evaluation Results



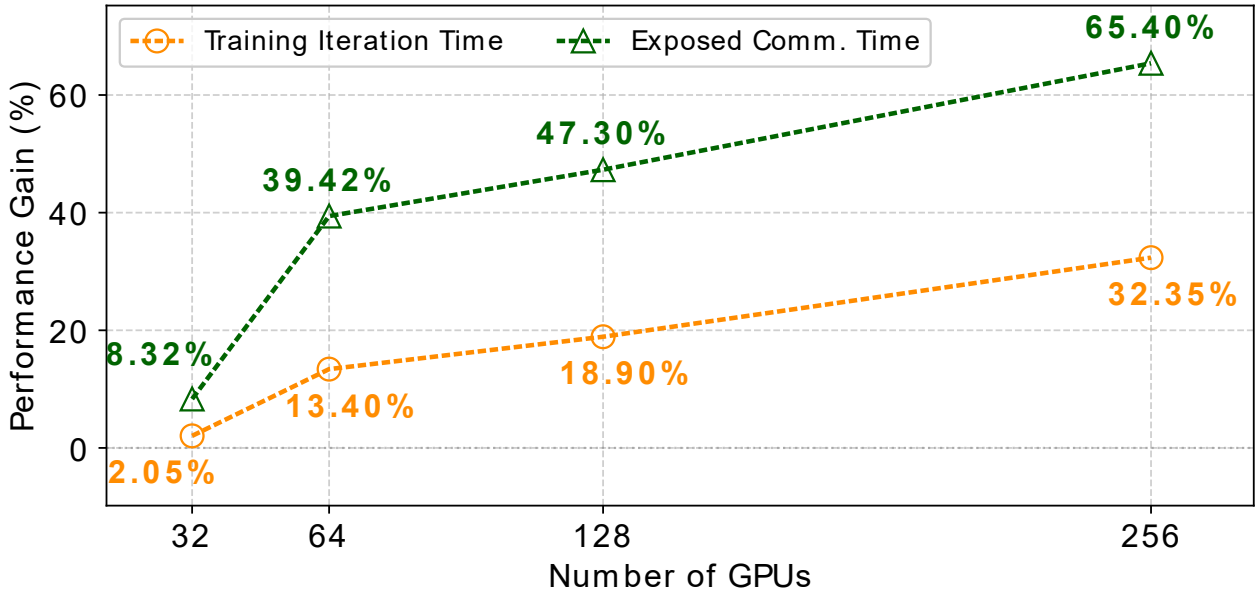
Only Inter-DC flows

Hierarchical exhibits a significantly **shorter tail FCT**!

! Hierarchical has **higher average FCT**!
In Ring AllReduce data is partitioned into chunks inversely proportional to the ring size:

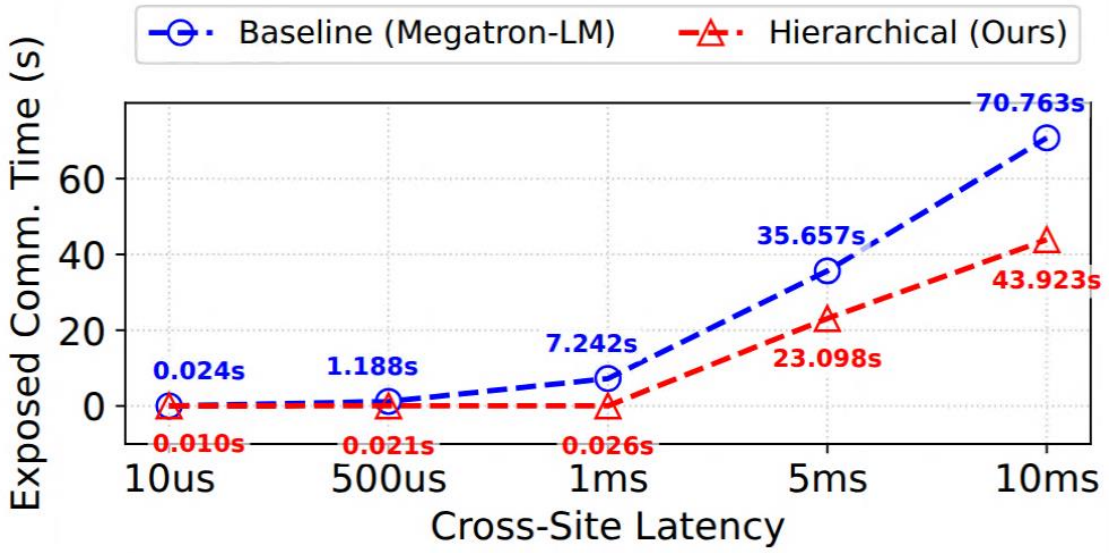
fewer participants = larger data partitions

Evaluation Results



We fix the inter-DC ring size to the number of leader replicas, **decoupling ring size from cluster size** and improving scalability

Evaluation Results



Hierarchical approach effectively **mitigates the impact of high inter-DC latencies**, maintaining better performance levels than the baseline strategy.

Conclusions and Future Work

The **hierarchical approach** with **leader replicas**:

- Reduces tail FCT and the overall training time
- Improves scalability with the number of GPUs
- Improves scalability with cross-site latency increase

Possibile **future directions**:

- Investigate scenarios with more than 2 DCs, heterogeneous clusters, PP > 1
- Analyze resilience via multiple leaders
- Test our approach on real hardware

Aggregate Local, Sync Global: A Hierarchical Approach to Efficient Geo-Distributed LLM Training

Francesco De Luca [*¶], **Francesco De Nadai** [*¶], Mariano Scazzariello [‡],
Tommaso Caiazzi [¶], Alireza Farshin [†], Marco Chiesa [§], Giuseppe Di Battista [¶]

[¶] Roma Tre University - [‡] RISE Research Institutes of Sweden - [†] NVIDIA - [§] KTH Royal Institute of Technology

NetCompute 2026 | Tokyo, Japan | May 18, 2026