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

***Abstract*—Nowadays, the exponential growth of LLMs in size, increasingly stringent data sovereignty regulations, and growing power and energy constraints in modern datacenters are driving the adoption of multi-datacenter infrastructures to distribute the training process. However, standard training frameworks prove ill-suited for these heterogeneous environments, lacking awareness of inter-datacenter constraints or stretching communication patterns across sites without differentiating between network tiers. This approach causes severe bottlenecks on high-latency inter-datacenter links, hindering scalability. To address this challenge, we propose a hierarchical communication strategy designed to decouple intra-site aggregation from global synchronization through the election of "leader model replicas". Experimental results show that the proposed hierarchical strategy reduces training iteration time by up to 32% and exposed communication time by up to 65% compared to the baseline Megatron-LM strategy, providing an effective solution for scaling training workloads across geographically distributed datacenters.**

***Index Terms*—Large Language Models, Datacenters, Multi-Site Training, Collective Communications Library.**

I. Introduction

LLM training pushes today's infrastructures to evolve at unprecedented speed. Recently, the development and rapid diffusion of Artificial Intelligence (AI), particularly within Machine Learning (ML), have intensified the demand for greater computing capacity. Large Language Models (LLMs), the current de-facto standard for a wide range of tasks thanks to their advanced capabilities [1], further amplify this trend. Training LLMs is a highly complex process characterized by extreme computational and communication challenges. Beyond requiring massive datasets and extensive hyperparameter tuning, LLM training relies on tightly coordinated distributed algorithms in which GPUs exchange large volumes of intermediate data at high frequency. As a result, scalability depends not only on the raw compute capacity but also on the efficiency of communication between devices and between nodes, which often becomes the dominant bottleneck at scale [2], [3]. LLM training workload characteristics have become a primary force driving architectural evolution in modern datacenters (DCs). Historically designed for end-user service delivery, DCs have progressively transformed into high-performance computing infrastructures capable of sustaining the massive data volumes and strict performance requirements of large-scale AI models. Meeting these demands has accelerated deep changes across multiple architectural layers. In particular, the widespread adoption of GPUs and highly parallel computing models has become essential to support the throughput and latency requirements of LLM training.

Physical and operational constraints limit single-site LLM training. LLMs continue to advance rapidly, with improvements in reasoning and overall capability driving unprecedented growth in computational requirements. Training state-of-the-art models now involves clusters composed of tens to hundreds of thousands of GPUs; *e.g.,* Grok 3 was reportedly trained on the Colossus supercomputer using a cluster with $\sim$200 K GPUs [4]. Future models are expected to demand even greater resources in the ongoing pursuit of Artificial General Intelligence (AGI). Accommodating this scale of growth within a single DC site is increasingly infeasible, as it is constrained by fundamental limits in power provisioning, thermal dissipation, and environmental sustainability [5]. In addition to infrastructure constraints, data sovereignty requirements increasingly limit the feasibility of single-site training. As models are trained on growing volumes of sensitive data, legal, privacy, and organizational policies may restrict data movement across regions [6], [7]. These constraints require training across distributed sites, even when sufficient compute capacity exists at a single site.

Communication, not computation, becomes the primary scalability bottleneck in multi-DC training. To overcome physical and operational constraints, an emerging approach is to distribute LLM training across multiple geographically distributed DCs [8]. While this strategy enables continued scaling and more flexible resource utilization, it also introduces new challenges. In particular, inter-DC communication becomes a dominant factor in determining end-to-end training performance. At large scales, distributed LLM training is increasingly communication-bound, making network performance a first-order architectural concern [9].

*These authors contributed equally to this work.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Current parallelism strategies and training frameworks fundamentally limit multi-DC LLM training. Addressing cross-site communication challenges requires rethinking the parallelism strategies used during training. Parallelism techniques (*e.g.,* tensor, pipeline, and data parallelism) define how computation and data are partitioned across devices and, consequently, how communication is structured and scheduled [10]. In multi-DC environments, these strategies must be adapted to account for heterogeneous network topologies [11], [12]. Traditional approaches, which are effective within a single DC, often assume a uniform network. For example, standard collective communication algorithms, *e.g.,* ring-based schemes, typically schedule uniform data transfers across all links, without distinguishing between high-bandwidth intra-node interconnects (*e.g.,* NVLink [13]) and significantly slower inter-node or inter-DC connections. As a result, naïve parallelism choices can amplify communication overheads and severely limit scalability in geographically distributed settings. State-of-the-art collective communication libraries (CCLs) [14]–[17] and training frameworks [10], [18] are primarily designed and evaluated for single-DC environments, where network latency and bandwidth are homogeneous. As a result, these libraries are largely agnostic to the heterogeneous nature of communication in multi-DC deployments. This lack of awareness exacerbates synchronization delays, leading to pronounced *straggler* effects in which GPUs remain idle while waiting for delayed parameter exchanges across slower links. Consequently, existing systems fail to scale in geographically distributed training scenarios.

Scope. In this preliminary work, we investigate communication and parallelism strategies for LLM training across multiple geographically distributed DCs. Specifically, we design a hierarchical synchronization mechanism that exploits locality and network hierarchy to overcome the scalability limitations of existing single-DC-oriented approaches. Our goal is to prioritize high-bandwidth intra-node and intra-DC communication while minimizing and isolating inter-DC traffic, restricting cross-DC links primarily to synchronization between different model replicas. By aligning parallelism mechanisms with the underlying network hierarchy, we aim to enable efficient multi-DC LLM training.

Contributions. We make three main contributions:

- We propose a hierarchical data parallelism for synchronizing different model replicas across geo-distributed DCs to mitigate high network latency;
- We benchmark our approach against Megatron-LM [10] through simulations, demonstrating up to a 32% reduction in per-iteration training time in multi-DC scenarios;
- We perform a scalability analysis, evaluating the communication overhead as the number of GPUs increases, confirming the robustness of our approach at scale.

II. Motivations & Background

We now discuss why existing LLM training approaches are inadequate in geographically distributed settings. We consider

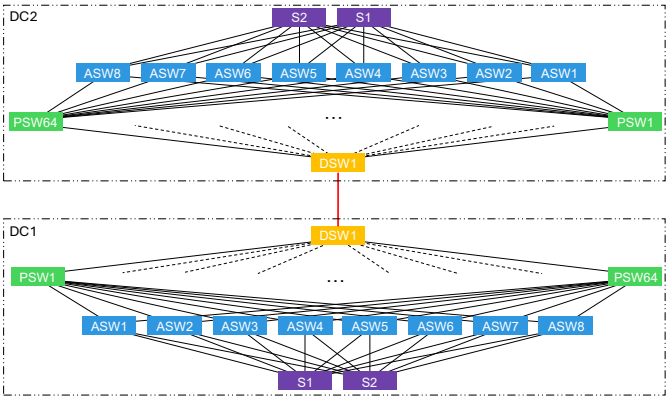


Fig. 1: Example of a multi-site topology for LLM training.

organizations (*e.g.,* financial or healthcare institutions) that seek to train proprietary LLMs using sensible data that reside in multiple geographically separated DCs. Increasingly strict data sovereignty regulations prohibit moving these datasets into a centralized data lake, making multi-site training a legal necessity rather than an optimization choice [6], [7]. Accordingly, we focus on sites that operate proprietary infrastructure with only tens to a few hundred GPUs. As a result, standard data parallelism, which assumes unrestricted sharding of data across compute nodes, must be fundamentally reconsidered to respect fixed physical and regulatory boundaries. This architecture aligns with cross-silo federated learning approaches [19], where stable, geographically distributed data silos collaboratively train a global model without exchanging data. However, unlike traditional federated learning, our focus is on high-performance training: the main challenge is efficiently synchronizing large model replicas over high-latency inter-site links.

Example scenario. We consider network topologies like the one shown in Fig. 1. Within each site, the fabric consists of servers equipped with 8 GPUs and 8 NICs [20], interconnected using a rail-aligned topology that has been shown to be highly efficient for LLM training [21]. Unlike single-site deployments, each spine switch (denoted as PSW in the figure) is additionally connected to an exit switch (denoted as DSW in the figure). The DSW is a commodity DC switch with buffering capacity in the order of 60 MB.[1] The two sites are interconnected by a single long-haul link (shown in red), whose latency can range from tens of microseconds to several milliseconds. We further assume a standard Ethernet fabric and that training relies on RDMA [22] to efficiently exchange intermediate results directly between GPUs, without CPU involvement. To address the challenges of inter-DC deployments, we assume a lossy RDMA transport. Under this design, the DSW does not generate PFC packets, which removes the risk of PFC storms [22], and instead directly drops packets when the buffer is full.

[1]In this scenario, carrier-grade switches with tens of GBs of buffer offer theoretical benefits. However, operators typically favor commodity switches to prioritize low hardware latency and seamlessly integrate with standard datacenter designs.

Inter-DC communication limits the scalability of distributed LLM training. Prior work [11] showed that inter-DC communication, particularly the synchronization of weight gradients across model replicas, is a major contributor to overall training time. Due to the high latency and limited bandwidth of long-haul links, efficient management of cross-DC traffic is essential to prevent excessive queuing in switch buffers. In our setting, funneling cross-DC traffic through a single, buffer-limited exit switch leads to packet buildup and severe congestion in the absence of further optimizations. This congestion can result in packet drops under lossy RDMA or trigger PFC in lossless configurations [22], ultimately disrupting throughput and degrading training performance.

Existing collective libraries lack the architectural awareness to handle the cross-site network. Existing CCLs, including TACCL [14], TE-CCL [15], SyCCL [16], and NCCL [17], aim to synthesize optimized collective algorithms by generating efficient execution schedules or exploiting topology-aware primitives. However, these designs rely on homogeneous network characteristics, restricting their optimizations and evaluations to single-site deployments. Moreover, these libraries encounter hard scalability limits even within a single DC (*e.g.,* 128 GPUs for TACCL and 256 GPUs for TE-CCL), making them inherently unsuitable for multi-DC LLM training. Recent versions of NCCL have begun incorporating topology awareness to better support multi-site deployments [23]. Similarly, higher-level training frameworks such as NVIDIA NeMo [18] have begun to explore hierarchical strategies to mitigate long-haul latency by prioritizing intra-DC communication before global synchronization [24]. However, these optimizations are *currently not implemented* in the standard communication backend. As a result, collective operations still treat the entire network as a single-site topology, significantly limiting their effectiveness in geographically distributed settings and leaving the underlying communication challenges largely unresolved.

Key insight: the theoretical advantages of hierarchical optimizations are agnostic to the specific CCL employed. In this paper, we argue that decoupling intra-DC aggregation from global synchronization constitutes a *fundamental* design principle that is independent of the specific algorithms implemented by CCLs. This separation enables a new optimization opportunity. Instead of relying on the rigid, flat topologies assumed by standard collectives, a hierarchical approach can explicitly orchestrate communication across levels. By treating the underlying CCL solely as a transport mechanism within local groups, this approach enables flexible, system-level optimizations that overcome the scalability limitations of individual library implementations.

III. Design

In this section, we first analyze the collective communication strategy employed by Megatron-LM [10]. We then introduce our hierarchical approach, which is designed to reduce communication overhead in multi-DC distributed training. To clearly illustrate the construction of logical rings and the resulting data flows, we consider a simplified reference topology with 32 GPUs distributed across two DCs (16 GPUs each), hosted on 8-GPU nodes. We further simplify the parallelization configuration by setting tensor parallelism $TP = 8$ (*i.e.,* sharding individual model layers across GPUs within a node), pipeline parallelism $PP = 1$ (*i.e.,* partitioning the model into sequential stages executed on different GPU groups), and data parallelism $DP = 4$ (*i.e.,* replicating the model to train on distinct data shards). This configuration results in two model replicas per DC, each fitting within a single 8-GPU node.

The industry standard: Megatron-LM. The Megatron-LM strategy structures communication into two primary logical dimensions. The first aggregates GPUs responsible for a single model replica, comprising $TP \times PP$ contiguous GPUs. Effectively, this dimension addresses the communication required for TP and PP. Since these operations, particularly TP, are latency-sensitive and involve frequent small transfers, this logical dimension is typically mapped onto the highest-performance network tier, leveraging high-speed intra-node interconnects such as NVLink (via NVSwitch) to minimize communication overhead. The second dimension manages DP by synchronizing gradients across replicas. DP communication groups are formed by connecting corresponding GPU ranks across replicas, using a fixed stride equal to the replica size (*i.e.,* $TP \times PP$). In our example with $TP = 8$ and $PP = 1$, each replica occupies a single node and the stride is 8. As a result, each DP ring links GPUs with the same intra-replica rank across nodes, *e.g.,* GPUs 0, 8, 16, and 24 for rank 1, as shown in Fig. 2. Since these groups span multiple nodes and potentially distinct geographic locations, this dimension maps onto inter-node and inter-DC network tiers. While less frequent than TP exchanges, DP communications transfer significantly larger payloads and are bandwidth intensive, relying on the scale-out fabric rather than local interconnects. Consequently, Megatron-LM maintains separate data-parallel rings, one for each GPU rank within a model replica.

Unlocking efficiency: a hierarchical approach. Our approach is designed to minimize the impact of long-haul link latency on site-local communications. To achieve this, we implement a hierarchical aggregation strategy that decouples the DP communication into two distinct tiers, intra- and inter-DC. To this end, we designate a *leader model replica* within each DC. GPUs belonging to the leader replica are solely responsible for synchronizing gradients across sites, while all non-leader replicas participate only in intra-DC communica-

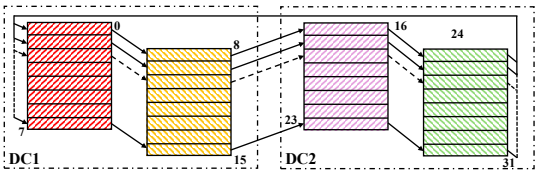


Fig. 2: Multi-DC Megatron-LM ring topology for DP. Distinct filling patterns denote separate model replicas, while arrows indicate the communication path across global ranks.

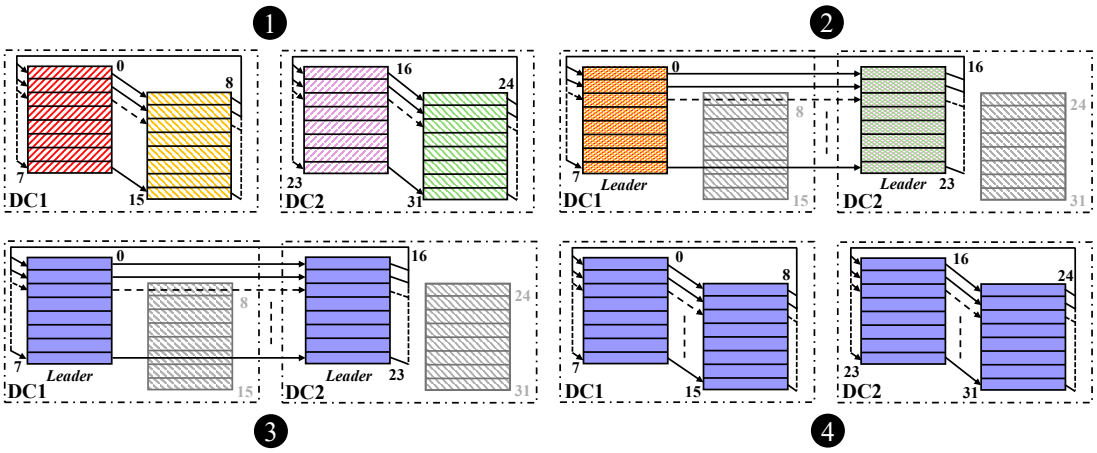


Fig. 3: Schematic overview of our hierarchical approach based on leader model replicas.

tion. This organization decouples the global baseline rings into multiple, independent local rings and a single global bridge that connects the different DCs. The synchronization process, shown in Fig. 3, consists of three stages. First, within each DC, all model replicas ❶ locally synchronize their gradients through an All-Reduce operation, where each GPU exchanges data with the corresponding GPUs of the other replicas in the same site. This step ensures that every GPU holds the locally aggregated gradient for its assigned parameter shard. Second, only the GPUs belonging to the designated leader replica participate in cross-site communication, performing ❷ an All-Reduce with their counterparts in the other DCs over the long-haul link. This step ❸ aggregates the locally reduced gradients into a global gradient. Finally, within each DC, the leader replica ❹ broadcasts the globally aggregated gradients to the non-leader replicas. Once the broadcast completes, all model replicas are fully synchronized, and the next training iteration can proceed. The key benefit of our approach is the clear separation of communication paths. Site-local replica synchronization can fully exploit low-latency, high-bandwidth interconnects, while inter-DC links are used only for global synchronization across sites. Crucially, this approach affects neither the number nor the size of cross-DC flows but effectively limits their interference with site-local traffic, confining them to a dedicated communication stage.

IV. Evaluation

This section evaluates the performance of our proposed communication approach against the Megatron-LM-based strategy, offering a preliminary validation aimed at addressing three fundamental questions:

Q1 *Can the hierarchical approach outperform the standard Megatron-LM-based strategy?*

Q2 *Does the hierarchical strategy sustain its benefits as the system scales?*

Q3 *Is the hierarchical approach robust to variations in inter-DC link latency?*

Experimental setup. We conducted our experiments using SimAI [25], a full-stack simulator designed to model the entire training stack. We extended SimAI to support multi-DC environments and implemented the Selective Repeat mechanism of Mellanox NICs to enable lossy RDMA, allowing the simulator to accurately model packet losses caused by inter-DC link limitations and switch buffer saturation. All evaluated topologies follow the same rail-aligned architecture shown in Fig. 1. We conducted simulations under four topology configurations, all assuming intra-DC links with 400 Gbps bandwidth and 1 μs latency, and a single inter-DC connection with 400 Gbps bandwidth and 500 μs latency, corresponding to an approximate physical distance of 100 km between the two exit switches. The evaluated configurations consisted of (i) 32 GPUs (16 GPUs per site), (ii) 64 GPUs (32 GPUs per site), (iii) 128 GPUs (64 GPUs per site), and (iv) 256 GPUs (128 GPUs per site).[2] All GPUs are modeled as NVIDIA H100 [26]. Workloads were generated using the SimAI Workload Generator component. We extended this module to incorporate the specific hierarchical communication primitives required by our proposed strategy. To ensure consistency, we enforce the use of identical workload traces across all simulation runs, where each run corresponds to a single training iteration consisting of a complete forward and backward propagation sequence.

Evaluation metrics. To quantify the benefits of our hierarchical approach, we rely on two key metrics. First, we analyze

[2]In future work, we plan to evaluate larger-scale configurations, increasing both the number of GPUs and the number of DCs involved.

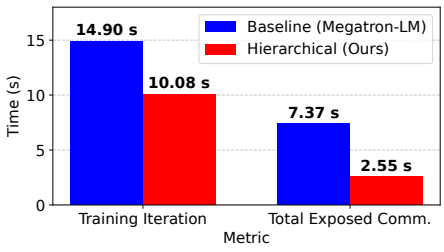
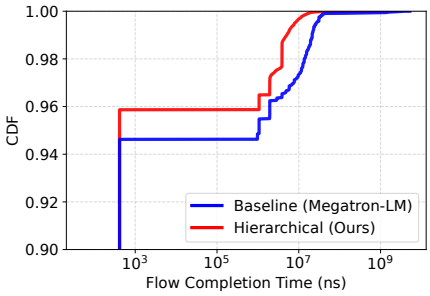
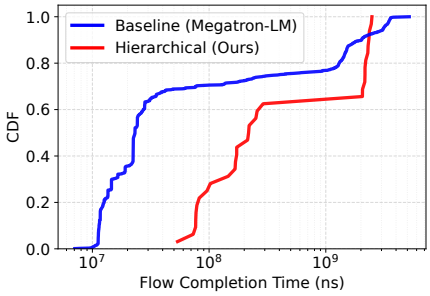


Fig. 4: Impact of the hierarchical strategy on training performance for a single iteration on the 256-GPUs topology.



(a) All flows (top 10%).



(b) Inter-DC flows.

Fig. 5: CDF for FCT.

the *training iteration time*, decomposing it into *computation* and *exposed communication time* (*i.e.,* the communication overhead not hidden by computation overlap). This breakdown provides a direct assessment of improved training efficiency. Second, we examine the Flow Completion Time (FCT) to understand network behavior and the reduction of tail latency.

Q1: The hierarchical approach reduces end-to-end training time by minimizing exposed communication overhead. We evaluate the hierarchical approach against the Megatron-LM-based strategy using the 256-GPUs topology (128 GPUs per site). As illustrated in Fig. 4, the hierarchical approach yields a substantial performance gain: the training iteration time decreases from 14.90 s to 10.08 s, marking a 32% reduction. Crucially, this improvement is driven by the optimization of the exposed communication time, which drops from 7.37 s to 2.55 s, decreasing the communication overhead by 65.4%. FCT analysis further corroborates these benefits. Specifically, the tail FCT drops from 5.19 s to 2.49 s, representing a 52% reduction. The CDFs of all flows and inter-DC flows, illustrated in Fig. 5a and 5b, respectively, clearly visualize this trend: the baseline exhibits a significantly longer tail compared to the more compact distribution of the hierarchical scenario. This reduction is critical as tail latency dictates the overall communication overhead and acts as the primary bottleneck for the training process. Interestingly, the CDFs also reveal that the hierarchical approach exhibits a higher average FCT compared to the baseline. This behavior stems from the mechanics of the Ring All-Reduce, where data is partitioned into chunks inversely proportional to the ring size: fewer participants imply larger data partitions. Our approach fixes the inter-DC ring size to two (*i.e.,* the number of DCs), producing larger chunks and therefore longer individual flow

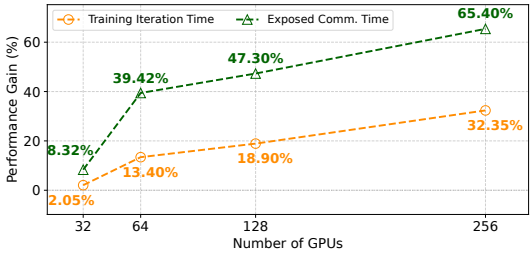


Fig. 6: Performance gain analysis scaling with varying GPUs.

durations. However, by splitting the global baseline rings into distinct intra- and inter-DC loops, our design mitigates the impact of long-haul link latency on local communications, leading to substantially lower tail latency.

Q2: The hierarchical strategy decouples inter-DC communication from cluster scale. A critical advantage of the hierarchical approach is its ability to decouple inter-DC traffic from the cluster size. To mitigate latency accumulation, we fix the inter-DC ring size to the number of sites, preventing the high inter-site latency from compounding across all the GPUs involved. This effectively reduces the communication overhead that impacts the training time, thereby minimizing GPU idle time. To evaluate this effect, we vary the number of GPUs per DC from 16 to 128 and measure the duration of a single training iteration. Fig. 6 illustrates the scalability of our approach, showing that the performance advantage of the hierarchical strategy over the baseline increases as the system scales to larger GPU counts. We plan to evaluate scenarios with more than two DCs as future work.

Q3: The hierarchical approach remains effective across different cross-site latencies. We wonder whether the performance gains of the hierarchical approach depend on specific latency regimes or remain robust across a range of cross-site delays. To assess this, we evaluate the impact of cross-site link latency on the overall exposed communication time. This analysis utilizes a fixed topology of 64 GPUs (32 GPUs per site), sweeping the inter-DC latency across values representative of real-world cross-DC training infrastructures [11]:

- Same-Campus Clusters: 10 μs;
- Cross-Campus Clusters: 100 μs;
- High-Latency Cross-Campus Clusters: 500 μs; and

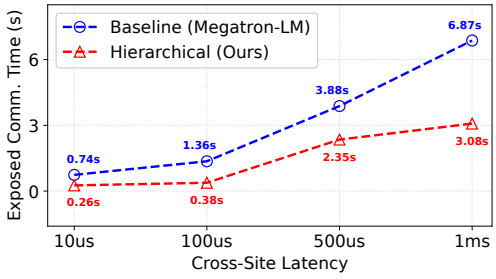


Fig. 7: Impact of increasing network latencies on exposed communication time.

- Same-Region Cloud: 1 ms;

Fig. 7 highlights that by constraining the size of inter-DC rings, the hierarchical approach effectively mitigates the impact of high inter-DC latencies, maintaining better performance levels than the baseline strategy.

V. Discussion

Thriving in asymmetry: conquering heterogeneous clusters. A major limitation of standard multi-site training is its inability to efficiently exploit asymmetric deployments. When DCs provide unequal compute capacity, synchronous DP training is bottlenecked by the smallest site, forcing bigger sites to idle while waiting for global synchronization. In contrast, our hierarchical approach naturally accommodates such heterogeneity through the election of a leader model replica. By confining cross-DC synchronization to the leader replicas, the system is robust to asymmetric deployments, allowing different DCs to host varying numbers of GPUs and, consequently, different numbers of model replicas. Unlike standard approaches, which typically assume symmetric configurations, this flexibility enables full utilization of available GPUs at each site, avoiding idle resources and improving both cost efficiency and overall system utilization. We plan to investigate asymmetric configurations as future work.

Compatibility with pipeline parallelism. While PP was not experimentally evaluated due to constraints within SimAI, our hierarchical approach is designed to be agnostic to the intra-replica parallelization strategy. Specifically, the election of a leader is performed at the *logical model replica level*. Consequently, our approach naturally encompasses all nodes constituting the leading replica, effectively supporting arbitrary pipeline depths. We plan to extend and evaluate our approach with PP in future work.

Resilience via multiple leaders. The proposed hierarchical leader-based strategy can be readily extended to enhance system resilience. Instead of relying on a single leader replica, the topology can support the election of a pool of backup leaders within each site, maintained in a *hot spare* configuration. This redundancy ensures that in the event of failure of the node(s) hosting a leader replica, an alternative replica can immediately assume the role of the inter-DC bridge. This mechanism prevents the halting of global synchronization, ensuring that the training process continues uninterrupted across DCs. We plan to investigate such scenario in future work.

VI. Conclusions

In this work, we proposed a novel hierarchical communication pattern that leverages leader-based replica selection to optimize data-parallel synchronization. Compared to the state-of-the-art Megatron-LM, our strategy yields a significant reduction in Exposed Communication Time of up to 65%. A fundamental strength of our approach is its scalability, as performance benefits become more pronounced as the system scales. Furthermore, our solution demonstrates robustness against inter-DC link latency variations. We believe that this approach represents a substantial contribution to the field, offering the potential to lower operational costs and minimize the overall environmental footprint. Future work will focus on evaluating the proposed method with asymmetric DCs and analyzing its interplay with pipeline parallelism. We will also investigate scenarios involving more than two DCs to further assess the effectiveness of our approach.

Acknowledgments

This work has been partially supported by Vinnova (the Sweden's Innovation Agency).

References

[1] W. X. Zhao *et al.*, "A Survey of Large Language Models," *arXiv preprint arXiv:2303.18223*, 2025.
[2] A. M. Gherghescu *et al.*, "A Look Into Training Large Language Models on Next Generation Datacenters," *arXiv preprint arXiv:2407.12819*, 2024.
[3] J. Fernandez *et al.*, "Hardware Scaling Trends and Diminishing Returns in Large-Scale Distributed Training," *arXiv preprint arXiv:2411.13055*, 2025.
[4] xAI. (2025) Colossus. [Online]. Available: https://x.ai/colossus
[5] United Nations Regional Information Centre (UNRIC). (2025) Artificial Intelligence: How much energy does AI use? [Online]. Available: https://unric.org/en/artificial-intelligence-how-much-energy-does-ai-use/
[6] D. K. Vohra. (2025) Why Europe's LLM Research Is Choosing Sovereign Cloud Infrastructure. [Online]. Available: https://www.nexgencloud.com/blog/thought-leadership/why-llm-research-in-europe-is-moving-to-sovereign-cloud-infrastructure
[7] R. Creemers *et al.*, "Translation: Cybersecurity Law of the People's Republic of China," *New America DigiChina*, June 2018.
[8] D. Narayanan *et al.*, "Efficient large-scale language model training on GPU clusters using megatron-LM," in *SC '21*. ACM, 2021.
[9] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.
[10] M. Shoeybi *et al.*, "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism," *arXiv preprint arXiv:1909.08053*, 2019.
[11] T. Chen *et al.*, "CrossPipe: towards optimal pipeline schedules for cross-datacenter training," in *ATC '25*. USENIX Association, 2025.
[12] S. Li *et al.*, "PyTorch distributed: experiences on accelerating data parallel training," *Proc. VLDB Endow.*, 2020.
[13] NVIDIA. (2025) NVLink and NVSwitch. [Online]. Available: https://www.nvidia.com/en-us/data-center/nvlink/
[14] A. Shah *et al.*, "TACCL: Guiding collective algorithm synthesis using communication sketches," in *NSDI 23*. USENIX Association, 2023.
[15] X. Liu *et al.*, "Rethinking Machine Learning Collective Communication as a Multi-Commodity Flow Problem," in *SIGCOMM '24*. ACM, 2024.
[16] J. Cao *et al.*, "Syccl: Exploiting symmetry for efficient collective communication scheduling," in *SIGCOMM '25*. ACM, 2025.
[17] NVIDIA. NVIDIA Collective Communication Library. [Online]. Available: https://github.com/NVIDIA/nccl
[18] O. Kuchaiev *et al.*, "NeMo: a toolkit for conversational AI and large language models," *arXiv preprint arXiv:1909.09577*, 2019.
[19] C. Huang *et al.*, "Cross-Silo Federated Learning: Challenges and Opportunities," *arXiv preprint arXiv:2206.12949*, 2022.
[20] NVIDIA. (2025) NVIDIA DGX B300. [Online]. Available: https://resources.nvidia.com/en-us-dgx-systems/dgx-b300-technical-brief
[21] K. Qian *et al.*, "Alibaba HPN: A Data Center Network for Large Language Model Training," in *SIGCOMM '24*. ACM, 2024.
[22] C. Guo *et al.*, "RDMA over Commodity Ethernet at Scale," in *SIGCOMM '16*. ACM, 2016.
[23] NVIDIA. (2025) NCCL Deep Dive: Cross-Data Center Communication and Network Topology Awareness. [Online]. Available: https://developer.nvidia.com/blog/nccl-deep-dive-cross-data-center-communication-and-network-topology-awareness/
[24] NVIDIA. (2024) Turbocharge LLM Training across Long-Haul Data Center Networks with NVIDIA NeMo Framework. [Online]. Available: https://developer.nvidia.com/blog/turbocharge-llm-training-across-long-haul-data-center-networks-with-nvidia-nemo-framework/

[25] X. Wang *et al.*, "SimAI: Unifying Architecture Design and Performance Tuning for Large-Scale Large Language Model Training with Scalability and Precision," in *NSDI 25*. USENIX Association, 2025.
[26] NVIDIA. (2022) NVIDIA H100 Tensor Core GPU Architecture. [Online]. Available: https://resources.nvidia.com/en-us-hopper-architecture/nvidia-h100-tensor-c