Simulating LLM training workloads for heterogeneous compute and network infrastructure

Sumit Kumar[1], Arjun Temura[1], Naman Sharma[1], Ramanjeet Singh[1], Meet Dadhania[2], Praveen Tammana[2], Satananda Burla[3], Abed Mohammad Kamaluddin[4], Rinku Shah[1]
[1]IIIT-Delhi, India [2]IIT Hyderabad, India [3]Marvell Technology Inc., USA [4]Marvell Technology Inc., India

Abstract

The growing demand for large-scale GPU clusters to train large language models (LLMs) poses a significant challenge to innovation due to high costs and limited accessibility. While state-of-the-art simulators address this issue, they assume a uniform infrastructure. However, device heterogeneity is unavoidable in cloud environments due to resource sharing, frequent updates in device generations, and the inherent intra-chip interconnect heterogeneity. We propose a heterogeneity-aware simulator for distributed LLM training that takes into account the real-world compute and network heterogeneity. Our simulator allows for custom configurations and models the impact of hardware diversity on training time.

CCS Concepts

• **Networks** → **Network performance modeling**; • **Computing methodologies** → **Simulation tools**.

Keywords

Distributed Training, LLM Simulator, Heterogeneous GPU Cluster

ACM Reference Format:
Sumit Kumar[1], Arjun Temura[1], Naman Sharma[1], Ramanjeet Singh[1], Meet Dadhania[2], Praveen Tammana[2], Satananda Burla[3], Abed Mohammad Kamaluddin[4], Rinku Shah[1]. 2025. Simulating LLM training workloads for heterogeneous compute and network infrastructure. In *2nd Workshop on Networks for AI Computing (NAIC '25), September 8–11, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3748273.3749212

1 Introduction and Motivation

In the past decade, the emergence of transformer models, commonly referred to as large language models (LLMs), such as GPT [26], Llama [29], and Mixtral [15], has revolutionized multi-task learning. With frequent LLM releases (fifteen per month on average [28]) and high training costs (10s of thousands of GPUs [17]), simulators are crucial for efficient cost and resource planning. The state-of-the-art simulators, ASTRA-sim [35] and SimAI [32], provide a full-system simulation framework for training clusters with homogeneous compute and networking infrastructure.

While companies like Meta [8] and Alibaba [25] use homogeneous GPU clusters, this model is increasingly unsustainable due to rapid hardware evolution [9], resource bottlenecks in shared clouds [13, 18, 34], and intra-chip interconnect bandwidth and latency variability in architectures such as the Grace-Hopper [7]. As a result, there is a growing need to utilize heterogeneous infrastructure for LLM training. Prior works [5, 14, 16, 24, 27, 30, 36–38] propose an optimal deployment plan for LLM training in a heterogeneous cluster comprising multiple GPU types and variable network bandwidth. However, these solutions are evaluated using costly real-world deployments [16] or oversimplified simulations [27]. Existing simulators [6, 10, 32, 35] lack support for key features that represent compute and network heterogeneity.

Our key idea. We propose to design a heterogeneity-aware distributed training simulator framework that extrapolates the LLM infrastructure and accurately predicts the performance of the custom deployment.

2 Challenges and Design Requirements

To optimize LLM training, state-of-the-art (SOTA) heterogeneity-aware LLM training solutions [14, 16, 24, 30, 33, 36, 37] explore ideas such as (a) non-uniform workload partitioning, based on compute capabilities, *e.g.*, the LLM model's MLP layer is compute-intensive and can be assigned to high compute GPUs for speedup [36], (b) heterogeneity-aware placement of distributed LLM model slices, *i.e.*, layers and tensors [37]. For example, leverage high bandwidth interconnects for model slices that communicate large amounts of data frequently, and (c) heterogeneity-aware training data sharding and orchestration, *e.g.*, in the case of multimodal data, image/video data must be trained on high-speed hardware [33]. We identify the gaps with the existing LLM training simulators, and present the abstractions (*A*), and components (*C*) needed for a heterogeneity-aware simulator design.

[A1] Abstractions for custom device groups and hybrid parallelism strategies. Existing simulators [32, 35] do not have the provision to specify custom device groups, *i.e.*, the set of heterogeneous compute devices that process a specific model slice (model partition or a set of model layers) or a training data slice. To enable non-uniform workload partitioning and heterogeneity-aware placement, the LLM training simulator must additionally support abstractions for: (a) heterogeneity-aware device groups across nodes, (b) custom parallelism configurations (Pipeline parallelism (PP), Tensor parallelism (TP), and Data parallelism (DP)) with variable batch sizes, and (c) flexible mapping of parallelism strategies to device groups.

[A2] Custom cluster and topology specification. The heterogeneity-aware simulator must provide abstractions to define diverse interconnects (*e.g.*, PCIe, NVswitch, and NVLink), bandwidth/latency parameters, and topological arrangements, enabling accurate simulation of diverse hardware infrastructures.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
NAIC '25, Coimbra, Portugal
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2082-6/2025/09
https://doi.org/10.1145/3748273.3749212

[C1] Non-uniform workload partitioning. The heterogeneity-aware simulator must: (a) distinguish between GPU types (*e.g.*, A100 vs. H100) and generate distinct workload traces tailored to the device group's role in the parallelism strategy, and (b) correctly simulate the non-uniform hybrid parallelism (*i.e.*, PP, TP, and DP) and the collective communication over custom device groups.

[C2] Resharding support for shape mismatch. Training across heterogeneous device groups with non-uniform parallelism configurations may result in tensor (or activation) shape mismatch during synchronization. Suppose Llama-2 (70B) [29] with 80 layers is trained on heterogeneous GPUs with two device groups (DG). DG_1 performs TP on 75 layers over 3 GPUs, and DG_2 performs TP on 5 layers over 2 GPUs. Due to the mismatch between the tensor shapes of DG_1 and DG_2, resharding is needed. The simulator must support automated resharding to adjust tensor shapes and ensure correctness in collective communication operations.

[C3] Heterogeneity-aware collective communication. Existing LLM training simulators [32] imitate NCCL [21] collective communication library optimizations. NCCL optimizes collective communication for: (a) efficient intra-node communication using bandwidth-aware graphs, (b) detects and selects efficient inter-node transport, and (c) maps logical ranks to physical devices to optimize performance. However, NCCL assumes GPU homogeneity and works only for NVIDIA GPUs. A heterogeneity-aware collective communication must support: (a) Graph generation for efficient collective communication in a heterogeneous cluster, *i.e.*, clusters with CPU-only nodes, or asymmetric architecture (*e.g.*, CPU+GPU+NPU), and (b) must be vendor agnostic.

[C4] Heterogeneous compute and interconnect simulation. The simulator must accurately measure and simulate the compute performance based on the bottleneck device in the ongoing transaction, and simulate custom network topology, interconnect capacities, and their delays.

3 Design and Preliminary Results

We extend the SimAI [32] simulation framework, which originally supports homogeneous GPU clusters and communication interconnects, to incorporate the abstractions and components discussed in §2. Figure 1 shows the primary components of our simulation framework with the preliminary results.

The input description component allows the user to feed in the custom heterogeneous configurations. The abstraction will allow the user to specify model parameters such as model dimensions, number of model layers, and training hyperparameters, framework parameters such as custom device groups, custom parallelism, mapping between them, and heterogeneous node and interconnect attributes such as latency and bandwidth.

Based on the input description, the workload generator profiles the workload layers on the specified device (*e.g.*, A100 and H100) using custom device group information to generate the custom workload file. We implemented a custom parser in SimAI's workload layer that parses a specific workload file for a given device group and registers the compute and communication events tied to specific operations.

The system layer is primarily responsible for logical resource management and scheduling. Using the framework parameters,

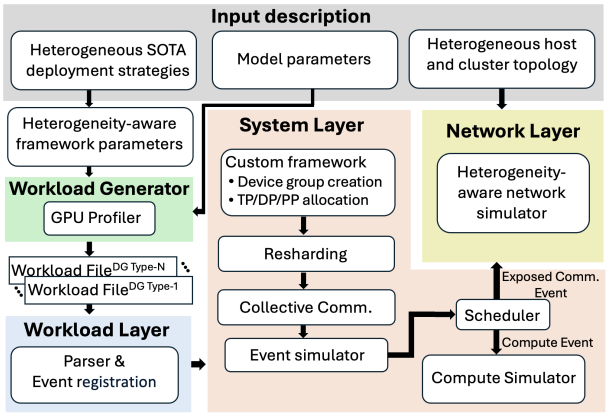

Figure 1: Design of heterogeneity-aware LLM training simulator.

we implemented custom device groups comprising homogeneous or heterogeneous GPUs for parallelism assignment. The device groups are then mapped to the hybrid of tensor, data, and pipeline parallelism based on the mentioned degree of parallelism. After parallelism mapping, the simulator will reshard the parameters and use custom channels to register the communication event for parameter synchronization. Then, the event simulator queues the registered events and logs all the registered events while simulating the distributed execution timeline using the scheduler. The scheduler coordinates the event stream between the compute and network simulators to ensure accurate modeling of event dependencies, resharding delays, and bandwidth contention. Our current implementation of the system layer only supports heterogeneous interconnect simulation.

We design a heterogeneity-aware network simulator engine on top of SimAI's ns3 module [2]. The engine uses the heterogeneous host and topology input description to instantiate and configure a network topology with custom interconnect latency, bandwidth, and processing delays, including intra-host connections and inter-host networks.

Setup. We set up the SimAI simulation framework [32] on a node with A100 and H100 GPUs. We generate the model workloads using the AICB workload benchmark [1], on A100 and H100 GPUs, to run simulations for homogeneous and heterogeneous configurations. We simulate heterogeneity over three LLM models, GPT-6.7B, GPT-13B, and Mixtral 8x7B with realistic training configurations [3, 4, 11, 19]. We simulate a rail-only topology [31], with each node having 8 GPUs and 8 RoCE NICs with Ampere [22, 23] Hopper [12, 20] configuration, with its bandwidth and delays.

Results. Our initial results show that a mixed Ampere and Hopper configuration (50%:50%) results in up to 9% higher flow completion time (FCT) for GPT-6.7B, a substantial 25.3× increase for GPT-13B, and a minimal 0.4% increase for Mixtral 8x7B compared to a homogeneous Ampere setup. Since collective communication is a blocking operation, the slowest flow dictates overall iteration time. These results emphasize the need for incorporating compute heterogeneity and applying scheduling optimizations such as those proposed in Metis [30] to mitigate tail latency in heterogeneous training environments.

References

[1] AICB. https://github.com/aliyun/aicb/tree/master Accessed: 2025-05-19.

[2] Songyuan Bai, Hao Zheng, Chen Tian, Xiaoliang Wang, Chang Liu, Xin Jin, Fu Xiao, Qiao Xiang, Wanchun Dou, and Guihai Chen. 2024. Unison: a parallel-efficient and user-transparent network simulation kernel. In *Proceedings of the Nineteenth European Conference on Computer Systems*. 115–131.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 https://arxiv.org/abs/2005.14165

[4] Yiming Cui and Xin Yao. 2024. Rethinking LLM Language Adaptation: A Case Study on Chinese Mixtral. arXiv:2403.01851 https://arxiv.org/abs/2403.01851

[5] Yifan Ding, Nicholas Botzer, and Tim Weninger. 2021. Hetseq: Distributed gpu training on heterogeneous infrastructure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15432–15438.

[6] Yicheng Feng, Yuetao Chen, Kaiwen Chen, Jingzong Li, Tianyuan Wu, Peng Cheng, Chuan Wu, Wei Wang, Tsung-Yi Ho, and Hong Xu. 2024. Echo: Simulating Distributed Training At Scale. *arXiv preprint arXiv:2412.12487* (2024).

[7] Luigi Fusco, Mikhail Khalilov, Marcin Chrapek, Giridhar Chukkapalli, Thomas Schulthess, and Torsten Hoefler. 2024. Understanding data movement in tightly coupled heterogeneous systems: A case study with the Grace Hopper superchip. *arXiv preprint arXiv:2408.11556* (2024).

[8] Adithya Gangidi, Rui Miao, Shengbao Zheng, Sai Jayesh Bondu, Guilherme Goes, Hany Morsy, Rohit Puri, Mohammad Riftadi, Ashmitha Jeevaraj Shetty, Jingyi Yang, et al. 2024. Rdma over ethernet for distributed training at meta scale. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 57–70.

[9] Amir Gholami, Zhewei Yao, Sehoon Kim, Coleman Hooper, Michael W Mahoney, and Kurt Keutzer. 2024. AI and memory wall. *IEEE Micro* (2024).

[10] Fei Gui, Kaihui Gao, Li Chen, Dan Li, Vincent Liu, Ran Zhang, Hongbing Yang, and Dian Xiong. 2025. Accelerating Design Space Exploration for {LLM} Training Systems with Multi-experiment Parallel Simulation. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. 473–488.

[11] Hugging Face. Mixtral. https://huggingface.co/docs/transformers/en/model_doc/mixtral. Accessed: 2025-05-21.

[12] Intel Corporation. Intel Ethernet Network Adapter E830-CQDA2 Specifications. https://www.intel.com/content/www/us/en/products/sku/239775/intel-ethernet-network-adapter-e830cqda2/specifications.html Accessed: 2025-05-22.

[13] Myeongjae Jeon, Shivaram Venkataraman, Amar Phanishayee, Junjie Qian, Wencong Xiao, and Fan Yang. 2019. Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. USENIX Association, Renton, WA, 947–960. https://www.usenix.org/conference/atc19/presentation/jeon

[14] Xianyan Jia, Le Jiang, Ang Wang, Wencong Xiao, Ziji Shi, Jie Zhang, Xinyuan Li, Langshi Chen, Yong Li, Zhen Zheng, et al. 2022. Whale: Efficient giant model training over heterogeneous {GPUs}. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. 673–688.

[15] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088* (2024).

[16] Yixuan Mei, Yonghao Zhuang, Xupeng Miao, Juncheng Yang, Zhihao Jia, and Rashmi Vinayak. 2025. Helix: Serving Large Language Models over Heterogeneous GPUs and Network via Max-Flow. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. 586–602.

[17] Meta AI. 2025. The LLaMA 4 Herd: The Beginning of a New Era of Natively Multimodal AI Innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/ Accessed: 2025-05-19.

[18] Zizhao Mo, Huanle Xu, and Chengzhong Xu. 2024. Heet: Accelerating elastic training in heterogeneous deep learning clusters. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 499–513.

[19] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient Large-Scale Language Model Training on GPU Clusters. *CoRR* abs/2104.04473 (2021). arXiv:2104.04473 https://arxiv.org/abs/2104.04473

[20] NVIDIA. NVIDIA H100 Tensor Core GPU. https://www.nvidia.com/en-in/data-center/h100/. Accessed: 2025-05-21.

[21] NVIDIA. 2024. NVIDIA Collective Communication Library (NCCL) Documentation. https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/index.html Accessed: 2025-05-19.

[22] NVIDIA Corp. NVIDIA A100 Tensor Core GPU. https://www.nvidia.com/en-in/data-center/a100/. Accessed: 2025-05-21.

[23] NVIDIA Corporation. ConnectX-6 Dx Datasheet. https://www.nvidia.com/en-in/networking/ethernet/connectx-6/. Accessed: 2025-05-21.

[24] Jay H Park, Gyeongchan Yun, M Yi Chang, Nguyen T Nguyen, Seungmin Lee, Jaesik Choi, Sam H Noh, and Young-ri Choi. 2020. {HetPipe}: Enabling large {DNN} training on (whimpy) heterogeneous {GPU} clusters through integration of pipelined model parallelism and data parallelism. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. 307–321.

[25] Kun Qian, Yongqing Xi, Jiamin Cao, Jiaqi Gao, Yichi Xu, Yu Guan, Binzhang Fu, Xuemei Shi, Fangbo Zhu, Rui Miao, et al. 2024. Alibaba hpn: A data center network for large language model training. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 691–706.

[26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* (2019).

[27] Foteini Strati, Zhendong Zhang, George Manos, Ixeia Sánchez Périz, Qinghao Hu, Tiancheng Chen, Berk Buzcu, Song Han, Pamela Delgado, and Ana Klimovic. 2025. Sailor: Automating Distributed Training over Dynamic, Heterogeneous, and Geo-distributed Clusters. arXiv:2504.17096 https://arxiv.org/abs/2504.17096

[28] Alan D. Thompson. Integrated AI: The Sky is Steadfast (2024 AI Retrospective). https://lifearchitect.ai/the-sky-is-steadfast/ Accessed: 2025-05-19.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[30] Taegeon Um, Byungsoo Oh, Minyoung Kang, Woo-Yeon Lee, Goeun Kim, Dongseob Kim, Youngtaek Kim, Mohd Muzzammil, and Myeongjae Jeon. 2024. Metis: Fast Automatic Distributed Training on Heterogeneous {GPUs}. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*. 563–578.

[31] Weiyang Wang, Manya Ghobadi, Kayvon Shakeri, Ying Zhang, and Naader Hasani. 2024. Rail-only: A low-cost high-performance network for training LLMs with trillion parameters. In *2024 IEEE Symposium on High-Performance Interconnects (HOTI)*. IEEE, 1–10.

[32] Xizheng Wang, Qingxu Li, Yichi Xu, Gang Lu, Dan Li, Li Chen, Heyang Zhou, Linkang Zheng, Sen Zhang, Yikai Zhu, et al. 2025. {SimAI}: Unifying Architecture Design and Performance Tuning for {Large-Scale} Large Language Model Training with Scalability and Precision. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. 541–558.

[33] Yujie Wang, Shenhan Zhu, Fangcheng Fu, Xupeng Miao, Jie Zhang, Juan Zhu, Fan Hong, Yong Li, and Bin Cui. 2025. Spindle: Efficient distributed training of multi-task large models via wavefront scheduling. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 1139–1155.

[34] Qizhen Weng, Wencong Xiao, Yinghao Yu, Wei Wang, Cheng Wang, Jian He, Yong Li, Liping Zhang, Wei Lin, and Yu Ding. 2022. MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. USENIX Association, Renton, WA, 945–960. https://www.usenix.org/conference/nsdi22/presentation/weng

[35] William Won, Taekyung Heo, Saeed Rashidi, Srinivas Sridharan, Sudarshan Srinivasan, and Tushar Krishna. 2023. Astra-sim2. 0: Modeling hierarchical networks and disaggregated systems for large-model training at scale. In *2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 283–294.

[36] Yongji Wu, Xueshen Liu, Shuowei Jin, Ceyu Xu, Feng Qian, Z Morley Mao, Matthew Lentz, Danyang Zhuo, and Ion Stoica. 2025. HeterMoE: Efficient Training of Mixture-of-Experts Models on Heterogeneous GPUs. *arXiv preprint arXiv:2504.03871* (2025).

[37] Ran Yan, Youhe Jiang, Xiaonan Nie, Fangcheng Fu, Bin Cui, and Binhang Yuan. 2025. HexiScale: Accommodating Large Language Model Training over Heterogeneous Environment. arXiv:2409.01143 https://arxiv.org/abs/2409.01143

[38] Fei Yang, Shuang Peng, Ning Sun, Fangyu Wang, Yuanyuan Wang, Fu Wu, Jiezhong Qiu, and Aimin Pan. 2024. Holmes: Towards distributed training across clusters with heterogeneous NIC environment. In *Proceedings of the 53rd International Conference on Parallel Processing*. 514–523.