BenchCouncil Transactions on Benchmarks, Standards and Evaluations 5 (2025) 100212

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Review Article

AICB: A benchmark for evaluating the communication subsystem of LLM training clusters

Xinyue Li *, Heyang Zhou, Qingxu Li, Sen Zhang, Gang Lu

Alibaba Cloud, Beijing, 100124, China

ARTICLE INFO

Keywords:
LLM training cluster
Benchmark
Collective communication
Distributed training

ABSTRACT

AICB (Artificial Intelligence Communication Benchmark) is a benchmark for evaluating the communication subsystem of GPU clusters, which includes representative workloads in the fields of Large Language Model (LLM) training. Guided by the theories and methodologies of Evaluatology, we simplified the real-workload LLM training systems through AICB that maintain good representativeness and usability. AICB bridges the gap between application benchmarks and microbenchmarks in the scope of LLM training. In addition, we constructed a new GPU-free evaluation system that helps researchers evaluate the communication system of the LLM training systems. To help the urgent demand on this evaluation subject, we open-source AICB and make it available at https://github.com/aliyun/aicb.

1. Introduction

The AI infrastructure is in rapid development with the flourishing of Artificial Intelligence [1,2]. For example, the explosion of the Large Language Model(LLM) applications leads to the fast evolution of the training frameworks [3–5], collective communication algorithms [6], network transports [7], and scale-out and scale-up network architectures [8]. Due to the large number of parameters in LLM, data is distributed across different GPUs for computation, requiring synchronization between these GPUs. Therefore, in LLM training, besides computation, communication also affects training efficiency. Consequently, evaluating the performance of the communication subsystem is a critical subject, that is, ensuring foundational technologies evolve in a manner that is both responsible and conducive to the continued progress in the field.

Some benchmarks are designed to evaluate the communication subsystem of a physical GPU cluster with high-performance scale-up and scale-out networks, including microbenchmarks and application benchmarks. However, the microbenchmarks are designed to evaluate the low-level peer-to-peer or collective communication operations under various message sizes and scales, while the application benchmarks only focus on the end-to-end performance. To bridge the gap, the community demands a new benchmark that produces workloads that mirror real-world LLM tasks, but focuses on the communication subsystem. In response to this demand, we built AICB and constructed the evaluation system using the methods proposed in [9].

The three essences of evaluating the communication subsystem of GPU clusters are as follows: (1) The Evaluation System (ES) is defined as a full-stack GPU cluster that LLM tasks can run. It includes the GPUs, the network infrastructure, and the software components running on it. The Evaluation Conditions should include all the capabilities and configurations of the hardware and software components that are tuned for the LLM training tasks that stakeholders concern. To be more specific, the Reference Evaluation System(RES) of AICB specifically targets the endpoint communication behavior through the end-to-end process of LLM training. (2) AICB provides measurement and testing tools that can generate and reproduce typical workloads in the Reference Evaluation System (RES) and ES. (3) The Value Function is the performance numbers output by AICB. It should give a clear quantified outcome of the comparison between different ECs, such as, the different collective algorithms, different parallel parameters, etc..

We construct the Pragmatic Evaluation System in two ways: (1) Rather than directly using all workloads from the real-world services, AICB is simplified to be more pragmatic. We elaborately select the workloads that can reflect the real-world behaviors with the criteria of spanning from the typical communication operations, message sizes, parallel parameters, optimization skills, and scales. (2) For researchers who lack GPUs, which is not uncommon in both industry and academia, we developed a GPU-free Evaluation System for the same evaluation subject. The NCCL(NVIDIA Collective Communication Library [6]) is hijacked to run on GPU-free cluster, but produces the same traffic.

* Corresponding author.
E-mail addresses: Lixinyue2019@bupt.edu.cn (X. Li), zhouheyang.zhy@alibaba-inc.com (H. Zhou), qingxu.lqx@alibaba-inc.com (Q. Li), zs411030@alibaba-inc.com (S. Zhang), yunding.lg@alibaba-inc.com (G. Lu).

https://doi.org/10.1016/j.tbench.2025.100212
Received 6 January 2025; Received in revised form 10 April 2025; Accepted 14 May 2025
Available online 2 June 2025

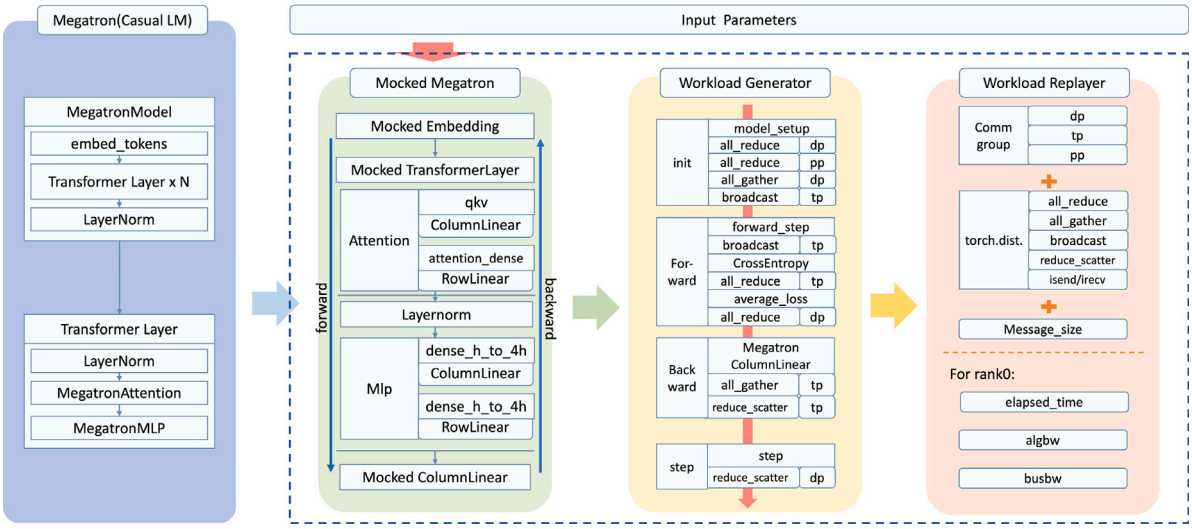

Fig. 1. Design of AICB with Megatron as an example. AICB gets the communication operation through Mocked Model, Workload Generator and Workload Replayer.

Meanwhile, the computation patterns are still kept in the evaluation, as we can collect them by running a specific computation tool on real GPU and afterwards they are embedded in the AICB workloads.

The main contributions of this paper are as follows:

(1) Guided by the principles of Evaluatology, we propose AICB, a benchmark to evaluate AI communication systems. By "hijacking" the training framework, we construct a pragmatic Evaluation System and develop a GPU-Free system through simulation.

(2) We use the end-to-end real elapsed time of every specific workload as metrics to evaluate the LLM communication subsystem. Through the assessment of case studies, we demonstrate the practicality of AICB in evaluating corresponding communication performance.

(3) Beyond the communication behavior, AICB can output the LLM training communication workload, serving as input for [10] to simulate the overall performance of the cluster training.

2. Related works

Existing benchmarks for evaluating physical GPU clusters can be mainly divided into microbenchmarks and application benchmarks. Microbenchmarks focus on assessing specific parts of the GPU cluster framework. For example, Nccl-test [11], developed by NVIDIA, is used to test and verify the performance and correctness of NCCL operations. It is specifically designed for NVIDIA GPUs and fully leverages the parallel processing capabilities. However, it has high dependencies on GPU hardware and CUDA environments. Perftest [12] provides a series of performance microbenchmarks based on Infiniband Verbs for hardware or software tuning and functional testing, but it requires specific hardware support.

Application benchmarks focus on evaluating the performance of model training. MLPerf [13] defines model architectures and training procedures for each benchmark, addressing ML evaluation challenges such as training randomness and significant time differences. AIBench [14] systematically refines and abstracts real-world application scenarios into scene, training, inference, micro, and synthetic AI benchmarks based on MLPerf. While these efforts have advanced ML training benchmark to some extent, there is a lack of overall focus on communication operations during LLM training.

AICB addresses this issue by providing precise evaluations of the communication subsystems. Instead of directly modifying these popular frameworks, AICB extract information through delicate monitoring tools and critical components.

3. AICB design

In the context of AI communication evaluation, a pragmatic composite evaluation system is needed to accurately represent common performance in AI training environments. When we design AICB, the communication system of LLM training is regarded as the evaluation subject with huge EC configurations, which is the input description module in AICB. The input contains a range of parameters to meet the expectations of stakeholders, such as different training models (e.g., GPT, LLaMA) with different scale of neural network, training configuration, popular training framework (e.g., Megatron, DeepSpeed [15]) with relative parallelism and aspects related to collective communication libraries like NCCL. These parameters can also be different Reference Evaluation Conditions(RECs) to compare AI training communication performance.

The core of AICB is implemented by "hijacking" the training framework. Fig. 1 illustrates the working principles of AICB using Megatron framework as an example to training models. Megatron is a highly scalable language model that improves training efficiency and speed through parallel processing of LLM. In practice, the Megatron structure starts from the input tokens, passing through multiple Transformer layer and norm layers, and ultimately reaching a linear layer to generate the model's output.

Instead of directly modifying frameworks, AICB extracts information through constructing critical components. Mocked Megatron is a simulated version of Megatron designed to simplify the complex model training process. Through the definitions of Mocked Embedding and Mocked Transformer Layer, it includes module of the Attention mechanism and the Multi-Layer Perceptron (MLP), implementing Column Linear and Row Linear calculations for each. Mocked Megatron approximates actual large model operations through these simulated components, aiming to create a simplified yet approximate training process model for subsequent communication system which enhancing the efficiency of simulating and testing communication patterns.

The primary purpose of the Workload Generator is to generate a list of communication workloads during the training process. It sequentially executes the training process according to the Mocked Model module. For example, during the Megatron training, it includes steps such as model initialization, forward propagation and backward propagation. During initialization, the focus is on model setup and data notification, with key operations including All-reduce, All-gather, and Broadcast, correspondingly choosing All-reduce for PP communication based on PP settings. Communication operations in model forward

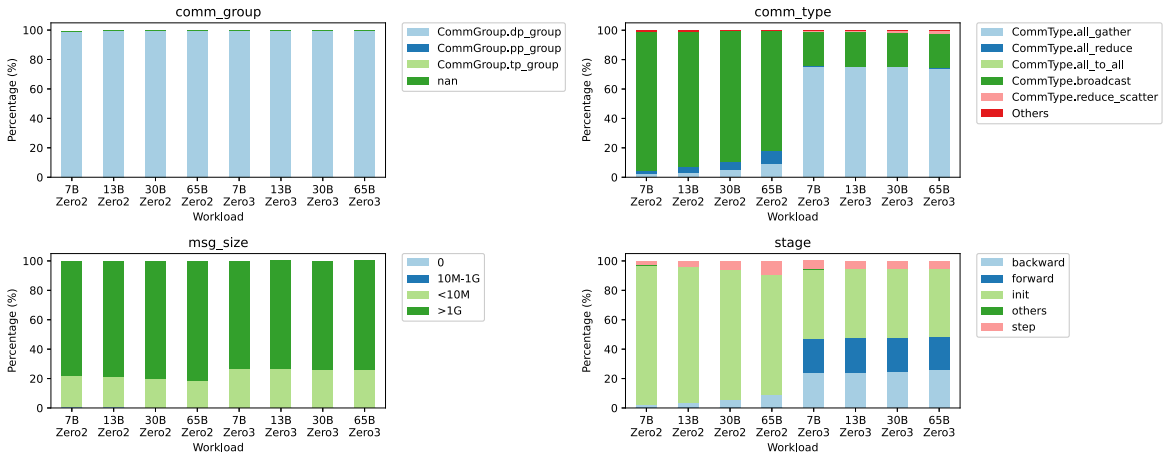

Fig. 2. Communication Distribution of LLaMA with under different scale and framework.

propagation mainly focus on TP, using All-reduce for DP communication when calculating loss. In the backward propagation stage, All-gather and Reduce-scatter update parameters among TP groups, and finally, during the step phase, DP synchronization prepares for the next round of training. Simulation of these steps gives a clear analysis of relations between each step of model training and communication groups with operations.

The task of the Workload Replayer is to apply the communication workload generated in the previous step to actual communication operations. By calling *torch.dist* with corresponding communication operations, it simulates the behavior of the workload, ensuring that the simulated communication load accurately reflects the communication overhead in the actual training process. Additionally, it measures communication operators, message sizes, and corresponding bandwidths on rank 0 of each step.

Notably, to more accurately replicate the communication assessment in LLM training, AICB offers an optional AIOB(Artificial Intelligence Operation Benchmark) mode, which is used to obtain the computation time for each operation of the actual model, accounting for the overlapping of multiple communication streams. Specifically, we break down the most computation-intensive parts, Attention and MLP, according to CUDA operations and extract the corresponding operations in the source code. This division not only facilitates the acquisition of computation time but also aids in the subsequent analysis of timing across different computational cores. Simulating multiple GPUs on a single GPU leverages the symmetry of GPU computation tasks and model training parameters are distilled to extract components that impact computation time. Parameters related to communication are used to slice the input calculation matrices and readjust weights according to the respective segments, simulating the model's division process. This ensures that the dimensions of the multiplied matrices during computation match those in real execution, allowing us to obtain accurate computation times for a partitioned model which is light weight but high accuracy.

4. Case study

4.1. Communication distribution

For the same cluster, AICB can be used to evaluate different distributions of AI training communication operation with composite ECs. Fig. 2 gives an example of the communication characteristics of LLaMA model with different model scales and parallelism strategies. In practice, DeepSpeed are mainly used to focus on data parallelism for synchronization, the DP-Group constitutes the majority within the communication group. Under ZeRO2, models have massive initialization stage which leads to an amount of broadcast operations for data notification for the first epoch. As model size increases, communication operations for the backward and step stage also increase, resulting in an increased reliance on All-gather and All-reduce. In ZeRO3, model parameters are integrated into the synchronization process, leading to a higher proportion of forward and backward operations compared to ZeRO2, with All-gather becoming the predominant communication method. In terms of message size, large traffic represents approximately 70%–80% and gradually increases with model scale. The distribution of communication operations provides a clear reflection of distributed frameworks in AI cluster training tasks and can be used to validate the communication differences of various RECs deployment.

4.2. Performance evaluation

We compared the workload generated by AICB with the actual training of the Megatron framework, integrating the data from communication groups, communication operation type and communication volume. Table 1 presents the communication results between AICB and realistic training. We tested GPT-7B under the Megatron framework with two A100 nodes, adopting TP = 8, PP = 1, DP = 2 as the parallel configuration. We gather and analyze communication characteristics of the workload generated by AICB and the actual Megatron training. It can be seen that both are quite similar in terms of communication features. Therefore, AICB's workload can represent the communication conditions of Megatron-GPT's actual training effectively, allowing AICB to be used for assessing the model's communication subsystem.

In addition to demonstrating the distribution of the communication subsystem, we have compiled models commonly used and selected those reflecting real-world LLM training workloads, forming a benchmark suite. We use elapsed time as an important baseline metric for evaluating communication system and output the detailed information for each specific communication collective operation.

In our experiments involving fixed collective communication operations, algorithm bandwidth is used to evaluate the performance of the cluster in Fig. 3. Similarly to the physical significance of other types of bandwidth, algorithm bandwidth is calculated based on the actual amount of data transmitted and the time required to complete these transmissions, as shown in (1). During the actual model training, the collective communication library selects the appropriate collective communication algorithm adaptively based on physical topology, communication patterns, and other relevant factors. Consequently, algorithm bandwidth can, to some extent, reflect the ability of the communication library to adapt its operations to the cluster. Higher algorithm bandwidth indicates that the communication library is able

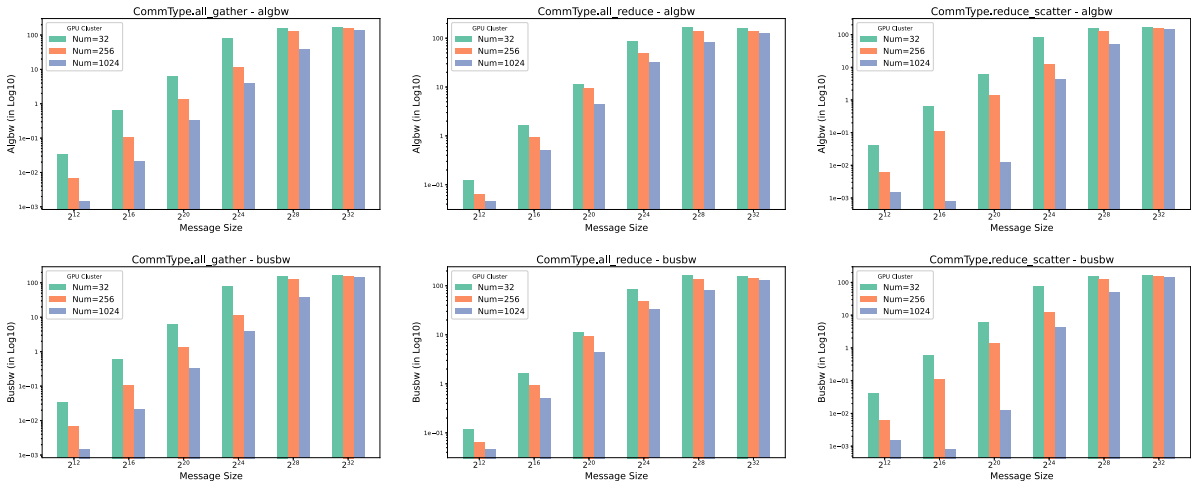

Fig. 3. Performance Evaluation for different cluster scale, message size and communication type.

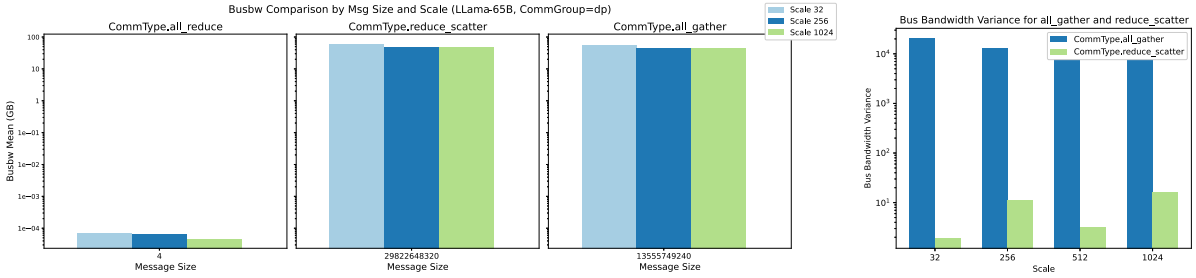

Fig. 4. Practical AICB simulation result of LLaMA 65B with different cluster scale.

Table 1
Comparison of communication between real training and AICB workload with megatron-GPT7B.

Comm type	Comm group	Real training		AICB workload	
		Message size	Number of comms	Message size	Number of comms
All-gather	dp-group	1.57 GB	10	1.55 GB	10
All-gather	tp-group	32 MB	33 280	32 MB	30 720
All-reduce	all	4B	10	4B	10
All-reduce	dp-group	4B	192	4B	160
All-reduce	tp-group	16 KB	576	16 KB	480
All-reduce	tp_group	3.03 MB	10	1 MB	10
All-reduce	tp-group	32 MB	192	32 MB	320
Reduce-scatter	dp-group	3.13 GB	10	3.09 GB	10
Reduce-scatter	tp-group	32 MB	22 688	32 MB	20 480

to utilize hardware resources more effectively, achieving more efficient data transmission.

$$\text{algbw (GB/s)} = \frac{\text{Size (GB)}}{\text{time (s)}} \tag{1}$$

$$\begin{cases} busbw_{\text{all\_reduce}} = algbw \cdot \dfrac{2(n-1)}{n} \\ busbw_{\text{all\_gather}} = algbw \cdot \dfrac{(n-1)}{n} \\ busbw_{\text{reduce\_scatter}} = algbw \cdot \dfrac{(n-1)}{n} \end{cases} \tag{2}$$

It is evident that as the size of the cluster increases, the value of algorithm bandwidth tends to decrease. To eliminate the influence of the number of GPUs on bandwidth, [11] introduces the concept of bus bandwidth, which serves as a metric to assess the efficiency of hardware utilization. This metric is derived by applying a specific calculation formula to the algorithm bandwidth, as shown in (2), to reflect the speed of inter-GPU communication irrespective of the cluster size, i.e., the number of GPUs used. By using this bus bandwidth, we can compare it against the hardware's theoretical peak bandwidth, thereby assessing the actual utilization efficiency of the hardware resources.

In the practical training simulation using the LLaMA65B model, we filtered the collective communication library operations corresponding to the DP group and the message sizes to evaluate the corresponding bus bandwidth. It is evident from Fig. 4 that as the cluster size increases, the bus bandwidth tends to decrease, aligning with the observed performance in Fig. 3. Due to the larger message volume, both the Reduce-scatter and All-gather operations generate higher bus bandwidth. We extracted and calculated the variance of the bus bandwidth for these two operations. It is clearly that the All-gather operation exhibits greater jitter. Intuitively, All-gather involves each participating process collecting data from all other processes, which entails a larger data volume and higher synchronization requirements. In contrast, Reduce-scatter performs partial reduction followed by the scattering of data, resulting in relatively lower synchronization demands and reduced pressure from network condition changes.

4.3. Workload for SimAI

SimAI [10] is a simulator we developed to evaluate complete GPU clusters, including components such as communication subsystems,

computing systems, and network architectures. The workload generated by AICB can serve as input for SimAI to simulate the conditions of model training, including various stages of model training, the size of communication data, communication operations, and the computation time corresponding to each stage. SimAI can form a comprehensive simulation evaluation system based on workload input, network topology information, and related network configurations, making it an important tool for evaluating large model infrastructure.

5. Conclusion

In this paper, we introduce AICB, a benchmark for evaluating the communication subsystem of LLM Training clusters. AICB focuses on communication subsystems in large-scale AI training clusters and defines appropriate ranges for RC to construct ES. By "hijacking" distributed training frameworks, it simulates specific collective communication operations. In addition to visualizing communication distribution, AICB uses bus bandwidth as a metric to evaluate the compatibility with specified clusters. AICB offers precise simulation and accurate evaluation of collective communications, providing substantial support for simulating and evaluating LLM training.

CRediT authorship contribution statement

Xinyue Li: Writing – review & editing, Writing – original draft, Visualization, Conceptualization. **Heyang Zhou:** Software, Conceptualization. **Qingxu Li:** Software, Conceptualization. **Sen Zhang:** Validation, Conceptualization. **Gang Lu:** Validation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Z. Jiang, H. Lin, Y. Zhong, Q. Huang, Y. Chen, Z. Zhang, Y. Peng, X. Li, C. Xie, S. Nong, Y. Jia, S. He, H. Chen, Z. Bai, Q. Hou, S. Yan, D. Zhou, Y. Sheng, Z. Jiang, H. Xu, H. Wei, Z. Zhang, P. Nie, L. Zou, S. Zhao, L. Xiang, Z. Liu, Z. Li, X. Jia, J. Ye, X. Jin, X. Liu, MegaScale: Scaling large language model training to more than 10,000 GPUs, 2024, arXiv:2402.15627.

[2] W. Li, X. Liu, Y. Li, Y. Jin, H. Tian, Z. Zhong, G. Liu, Y. Zhang, K. Chen, Understanding communication characteristics of distributed training, in: Proceedings of the 8th Asia-Pacific Workshop on Networking, APNet '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 1–8, http://dx.doi.org/10.1145/3663408.3663409.

[3] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, B. Catanzaro, Megatron-LM: Training multi-billion parameter language models using model parallelism, 2020, arXiv:1909.08053.

[4] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, A. Phanishayee, M. Zaharia, Efficient large-scale language model training on GPU clusters using megatron-LM, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21, Association for Computing Machinery, New York, NY, USA, 2021, http://dx.doi.org/10.1145/3458817.3476209.

[5] V. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, B. Catanzaro, Reducing activation recomputation in large transformer models, 2022, arXiv:2205.05198.

[6] NCCL, NVIDIA collective communications library (NCCL), 2024, https://developer.nvidia.com/nccl (Online; Accessed 4 October 2024).

[7] J. Zhang, Y. Wang, X. Zhong, M. Yu, H. Pan, Y. Zhang, Z. Guan, B. Che, Z. Wan, T. Pan, T. Huang, PACC: A proactive CNP generation scheme for datacenter networks, IEEE/ACM Trans. Netw. 32 (3) (2024) 2586–2599, http://dx.doi.org/10.1109/TNET.2024.3361771.

[8] K. Qian, Y. Xi, J. Cao, J. Gao, Y. Xu, Y. Guan, B. Fu, X. Shi, F. Zhu, R. Miao, C. Wang, P. Wang, P. Zhang, X. Zeng, E. Ruan, Z. Yao, E. Zhai, D. Cai, Alibaba HPN: A data center network for large language model training, in: Proceedings of the ACM SIGCOMM 2024 Conference, in: ACM SIGCOMM '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 691–706, http://dx.doi.org/10.1145/3651890.3672265.

[9] J. Zhan, L. Wang, W. Gao, H. Li, C. Wang, Y. Huang, Y. Li, Z. Yang, G. Kang, C. Luo, H. Ye, S. Dai, Z. Zhang, Evaluatology: The science and engineering of evaluation, BenchCouncil Trans. Benchmarks, Stand. Eval. 4 (1) (2024) 100162, http://dx.doi.org/10.1016/j.tbench.2024.100162, URL https://www.sciencedirect.com/science/article/pii/S2772485924000140.

[10] X. Wang, Q. Li, Y. Xu, G. Lu, D. Li, L. Chen, H. Zhou, L. Zheng, S. Zhang, Y. Zhu, et al., 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), 2025.

[11] Nccltests, NCCL-tests, 2024, https://github.com/NVIDIA/nccl-tests (Online; Accessed 4 October 2024).

[12] Perftest, Infiniband verbs performance tests, 2024, https://github.com/linux-rdma/perftest (Online; Accessed 4 October 2024).

[13] P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf, et al., Mlperf training benchmark, Proc. Mach. Learn. Syst. 2 (2020) 336–349.

[14] F. Tang, W. Gao, J. Zhan, C. Lan, X. Wen, L. Wang, C. Luo, Z. Cao, X. Xiong, Z. Jiang, T. Hao, F. Fan, F. Zhang, Y. Huang, J. Chen, M. Du, R. Ren, C. Zheng, D. Zheng, H. Tang, K. Zhan, B. Wang, D. Kong, M. Yu, C. Tan, H. Li, X. Tian, Y. Li, J. Shao, Z. Wang, X. Wang, J. Dai, H. Ye, Aibench training: Balanced industry-standard AI training benchmarking, in: 2021 IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS, 2021, pp. 24–35, http://dx.doi.org/10.1109/ISPASS51385.2021.00014.

[15] J. Rasley, S. Rajbhandari, O. Ruwase, Y. He, DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 3505–3506, http://dx.doi.org/10.1145/3394486.3406703.