

Performance Characterization of Large Language Models on High-Speed Interconnects

Hao Qi^{*}, Liuyao Dai^{*}, Weicon Chen^{*}, Zhen Jia[†], and Xiaoyi Lu^{*§}

^{*}University of California Merced, Merced, CA, USA

{hqi6, ldai8, wchen97, xiaoyi.lu}@ucmerced.edu

[†]Amazon Web Service, Santa Clara, CA, USA

zhej@amazon.com

Abstract—Large Language Models (LLMs) have recently gained significant popularity due to their ability to generate human-like text and perform a wide range of natural language processing tasks. Training these models usually requires a large amount of computational resources and is often done in a distributed manner. The use of high-speed interconnects can significantly influence the efficiency of distributed training. Therefore, there poses a need for systematic studies to explore the distributed training characteristics of these models on high-speed interconnects. This paper presents a comprehensive performance characterization of representative large language models: GPT, BERT, and T5. We evaluate their training performance in terms of iteration time, interconnect utilization, and scalability, over different high-speed interconnects and communication protocols, including TCP/IP, IPoIB, and RDMA. We observe that interconnects play a vital role in LLM training. Specifically, RDMA-100 Gbps outperforms IPoIB-100 Gbps and TCP/IP-10 Gbps by an average of 2.51x and 4.79x regarding training iteration time, and scores the highest interconnect utilization (up to 60 Gbps) in both strong and weak scaling, compared to IPoIB with up to 20 Gbps and TCP/IP with up to 9 Gbps, leading to the shortest training time. We also observe that larger models tend to have higher requirements for communication bandwidth, especially for AllReduce during backward propagation, which can take up to 91.12% of training time. Through our evaluation, we envision opportunities to improve the communication time for better training performance of LLMs. We extensively explore and summarize the role communication plays in distributed LLM training.

Index Terms—Large language models, Characterization, Transformer, GPT, BERT, T5

I. INTRODUCTION

Generative Artificial Intelligence (AI) is becoming a scorching topic recently. Transformer-based large language foundation models [1]–[5] have emerged as powerful techniques for various natural language processing (NLP) tasks, such as language translation, text generation, and sentiment analysis. These models have the remarkable ability to understand and generate human-like text, making them indispensable for various applications in industries such as healthcare [6], finance [7], and marketing [8]. However, obtaining effective LLMs is full of challenges due to their natural characteristics of immense parameter size and complexity, often consisting of millions, billions, or even trillions of parameters.

Training LLMs require substantial computational power and memory capacity to accommodate the vast model weights. As a result, LLM training is extremely resource-consuming, placing high demands on the underlying infrastructure [9]. Distributed training

is applied to address resource eagerness and accelerate the training process [10], which involves partitioning the model and/or the training data across multiple compute nodes (usually GPUs and high-speed interconnects equipped), allowing for parallel training and reducing the overall training time.

Distributed training for LLMs introduces new challenges, particularly regarding communication and coordination among the nodes and GPUs. This is where high-speed interconnects come into play. As indicated by Figure 1, high-speed interconnects (such as InfiniBand EDR/HDR/NDR, RoCEv2, high-speed Ethernet, etc.) play a vital role in facilitating efficient data transfer and synchronization during LLM training, since they are essential for achieving fast and scalable communication between nodes and GPUs, minimizing the communication overhead, and maximizing the overall system performance.

The sheer volume of training data and the need for distributed GPU-enabled training of LLMs further intensify the demand for high-performance interconnects, without which the communication overheads can quickly threshold the scalability and efficiency of LLM training. The past decades have witnessed the compute capability of modern HPC systems scaling at more than twice the pace of interconnect bandwidth across generations [11]. This trend raises myriad potential research problems for achieving efficient and scalable LLM training. Some of them include:

- ① Will high-speed interconnects become the bottleneck for communication, and what proportion of the training process is occupied by communication for various types and configurations of LLMs?
- ② Are the current high-performance interconnects utilized well during different distributed training scenarios?
- ③ What kind of quantitative performance impact will different networking technologies and protocols (such as RDMA, IPoIB, TCP/IP) have on various LLMs training?

To answer these questions, this paper explores and characterizes the influence and importance of high-performance interconnects in distributed training of various LLMs. By analyzing and evaluating the impact of different interconnect technologies and protocols, we seek to provide the research community with a better understanding of the significance of high-speed interconnects. To achieve this goal, our methodology focuses on several dimensions that are crucial in the context of LLM training:

- (1) **Workload:** We will consider representative LLM models under different configurations, such as GPT-2 [3] (Medium and Large),

§ is the corresponding author.

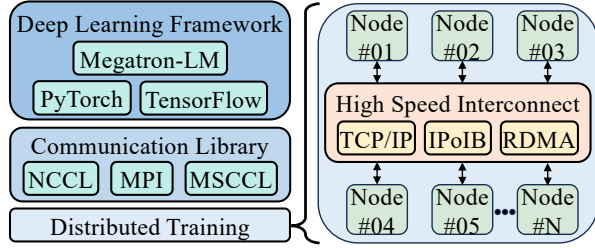


Fig. 1: Comm. and Interconnect in Distributed Training.

BERT [1] (Large), and T5 [12] (Large). Additionally, we will employ the widely used English Wikipedia dataset (enwiki) [13] as it represents a comprehensive and diverse corpus for training LLMs. **(2) Interconnect/Protocol:** We will evaluate and compare different interconnect options, including the widely used TCP/IP [14], IP over InfiniBand (IPoIB) [15], and the efficient Remote Direct Memory Access (RDMA) protocol [16]. Each networking technology offers varying performance and scalability characteristics, significantly impacting the overall training efficiency.

(3) Scalability: The ability to scale LLM training is critical for handling increasingly large models and datasets. We will investigate two types of scalability: strong scaling and weak scaling. Strong scaling measures the performance improvement achieved by increasing the compute resources on fixed problem sizes. In contrast, weak scaling evaluates the ability to maintain a constant workload per node/GPU while scaling out the overall system.

With our performance characterization, this paper makes the following key contributions:

- 1 We propose a performance characterization methodology for LLM on high-speed interconnects. It closely aligns with real-world distributed training for LLM, rather than simulated experimental settings. With this methodology, we measure key metrics such as communication latency, network bandwidth utilization, and training scalability to quantify the benefits and limitations of each interconnect/protocol option.
- 2 We systematically characterize the performance of LLM training on interconnects and make observations. Our results show that high-performance network protocols, like (GPUDirect) RDMA, can significantly outperform other protocols in training performance, such as IPoIB and TCP/IP, by 2.51x and 4.79x. Our characterization also shows that backward parameter sync time can take up to 91.72% in training time, implying communication can become the major bottleneck for distributed LLM training.
- 3 By thoroughly examining the dimensions in our methodology, we extensively explore and summarize the proportion of communication in the training process. We shed light on the role and significance of high-speed interconnects in distributed LLM training. This paper will serve as a valuable resource for the community by comprehensively characterizing the contribution of high-performance interconnects on LLM workloads.

II. OVERVIEW OF SELECTED LLMs

The prominent LLMs we choose to evaluate, including GPT [3]–[5], BERT [1], and T5 [12], have revolutionized natural language

processing with their large model sizes, sophisticated architectures, and exceptional performance on various NLP tasks. These models and their variants have opened up new possibilities for text generation, sentiment analysis, machine translation, and more applications by pushing the boundaries of language understanding and generation. The comparison of their architectures is shown in Figure 2.

A. GPT Overview

The GPT (Generative Pre-trained Transformer) is a family of transformer-based decoder models (Figure 2a) developed by OpenAI that are used in natural language processing to generate coherent and diverse texts on various topics. The GPT family consists of models like GPT-2, GPT-3, GPT-3.5, and the most recent GPT-4. These models have been trained on large datasets and can generate human-like text using various styles and topics. At the forefront of this family of models is ChatGPT, which is based on GPT-3.5. ChatGPT has gained tremendous popularity due to its ability to engage in interactive and human-like conversations. By leveraging the power of deep learning and advanced language modeling techniques, ChatGPT can produce contextually relevant responses and hold meaningful discussions with users. The GPT series has showcased remarkable performance across various natural language processing tasks. They excel in text summarization, i.e., distilling lengthy passages into concise summaries; answering questions, i.e., providing accurate and relevant answers based on the context; and text completion, i.e., generating coherent and contextually appropriate text to continue a given prompt.

B. BERT Overview

BERT (Bidirectional Encoder Representations from Transformers) is another influential LLM. BERT has gained significant attention in the NLP community due to its outstanding performance across multiple tasks. The model size can vary depending on the variants, with BERT-Base consisting of 110 million parameters and larger variants like BERT-Large having up to 340 million parameters. BERT’s bidirectional transformer encoder (Figure 2b) captures contextual information from both left and right contexts, enabling it to excel in tasks requiring a deep understanding of sentence-level semantics and contextual word representations. By leveraging the bidirectional nature of the model, BERT achieves remarkable results in tasks such as text classification, named entity recognition, sentiment analysis, natural language inference, and more. It has become a foundational model in the NLP landscape, driving advancements in various downstream applications and setting state-of-the-art performance on numerous benchmarks.

C. T5 Overview

T5 (Text-To-Text Transfer Transformer) is a transformative LLM known for its text-to-text transfer learning approach. T5 comes in different sizes, with the largest variant, T5-XXL, boasting 11 billion parameters. The architecture of T5 revolves around a transformer-based encoder-decoder model (Figure 2c). The encoder and decoder leverage transformer networks, enabling T5 to address a wide range of NLP tasks by framing them as text-to-text problems. This approach allows T5 to generalize across different tasks and domains, simplifying the training process and promoting

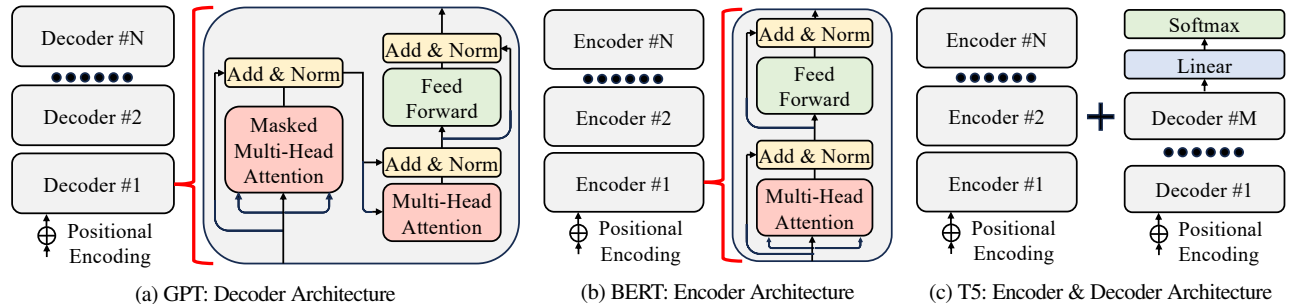


Fig. 2: Overview of Transformer-based LLMs' Architectures

knowledge transfer. T5 has demonstrated impressive performance across diverse applications, including text summarization, machine translation, question-answering, document classification, and more. Its strength lies in the transfer learning paradigm, where pre-training on large corpora allows fine-tuning for specific downstream tasks. The robustness and adaptability of T5 have contributed to its popularity among researchers and practitioners in the NLP community, facilitating advancements in various areas of language understanding and generation.

D. Summary

Large language models have significantly advanced the regime of natural language processing. We select three kinds of models representing popular types of transformer-based architectures: decoder models, encoder models, and encoder & decoder models. Such sophisticated architectures have enabled them to perform remarkably on various NLP benchmarks. We will explore how well these architectures accommodate different networking technologies and evaluate their performance implications.

III. CHARACTERIZATION METHODOLOGY

A. Methodology Overview

In this subsection, we present an overview of the methodology used to characterize the impact of high-performance interconnects on large language models. As mentioned in Section I, our characterization focuses on three pivotal dimensions: workload, interconnect/protocol, and scalability, as shown in Figure 3. By systematically examining these dimensions, we gain insights into the performance and efficiency of LLM training under different interconnect settings. The following paragraphs provide a brief explanation of each dimension:

1) *Workloads*: The choice of LLMs and datasets is crucial in our characterization methodology. We consider some popular open-source LLMs, including GPT-2-Medium, GPT-2-Large, BERT-Large, and T5-Large, representing a range of model sizes, architectures, and application domains. These models have been widely adopted in various NLP tasks and exhibit different computational requirements. Additionally, we select one representative dataset enwiki [13]. By evaluating the impact of high-performance interconnects across different LLMs, we assess high-speed interconnects' generalizability and performance characteristics in diverse contexts. More details for the workloads are summarized in Table I.

TABLE I: Comparison of Selected LLMs

| Model | Architecture | Layers | Hidden Size | Attention Head | Parameters |
|--------------|--------------|--------|-------------|----------------|------------|
| GPT-2 Medium | Decoder | 24 | 1024 | 16 | 345M |
| GPT-2 Large | Decoder | 36 | 1280 | 20 | 774M |
| BERT Large | Encoder | 24 | 1024 | 16 | 340M |
| T5 Large | En/Decoder | 24 | 1024 | 16 | 770M |

2) *Interconnect/Protocol*: The choice of interconnect and protocol is another essential aspect of our methodology. We consider different interconnect technologies such as TCP/IP, IPoIB, and RDMA (with GPUDirect). Each interconnect technology offers different features, performance characteristics, and levels of efficiency. By exploring the influence of these interconnect options on LLMs, we can understand how they affect communication patterns, data transfer rates, and overall performance. This analysis will provide insights into the suitability of specific interconnect technologies for LLM workloads.

3) *Scalability*: Scalability is an essential characteristic of assessing the efficiency and effectiveness of LLMs in distributed computing environments. In our methodology, we evaluate both strong scaling and weak scaling aspects. Strong scaling measures the performance improvement achieved by increasing the compute resources with a fixed problem size. In contrast, weak scaling evaluates the performance of LLMs when both the problem size and the number of compute resources scale proportionally. By examining the scalability of LLMs across different interconnect technologies under the data parallelism training architecture, we can identify potential bottlenecks, scalability limits, and the overall efficiency of the distributed training process.

By considering these three dimensions in our methodology, we comprehensively assess the influence of high-performance interconnects on LLMs. This characterization will help us gain valuable insights into the interplay among interconnect technologies, scaling scenarios, and the performance of LLM workloads.

B. Frameworks and Dataset

1) *Framework*: We leverage the Megatron-LM [10] framework for our characterization methodology as our primary distributed training framework. Megatron-LM is a powerful framework

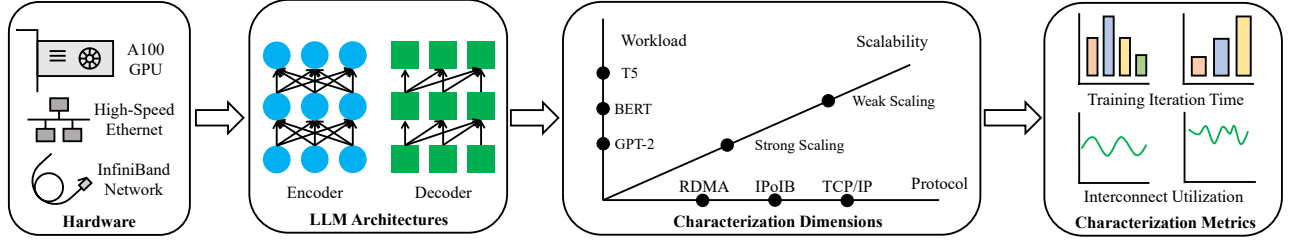


Fig. 3: Characterization Methodology.

specifically designed for training large-scale language models. It provides efficient and scalable implementations of distributed training algorithms, making it an ideal choice for our investigation. Megatron-LM offers support for various interconnect technologies, enabling us to effectively explore the impact of high-performance interconnects on LLMs. By utilizing Megatron-LM, we can ensure the consistency and reliability of our experimental setups and facilitate meaningful comparisons across different interconnects.

2) *Dataset*: In our characterization methodology, we utilize the enwiki [13] dataset as a representative example of a large-scale dataset. The enwiki dataset (20.4 GB) is derived from English Wikipedia and contains vast text documents spanning diverse topics and genres. It provides a rich and challenging corpus for training and evaluating LLMs. By incorporating the enwiki dataset into our experiments, we can assess the performance and scalability of LLMs under the influence of high-performance hot interconnects using a real-world, content-rich dataset. The size and complexity of the enwiki dataset enable us to explore the implications of interconnect technologies and parallelization strategies in handling large-scale language modeling tasks.

IV. PERFORMANCE EVALUATION

A. Experimental Setup

Pinnacles Cluster [17]: The NSF-funded Pinnacles cluster at UC Merced is equipped with 8 GPU nodes. Each node has two Intel 28-Core Xeon Gold 6330 CPUs (2.0GHz), 256GB DRAM, 2x NVIDIA Tesla A100 40GB GPUs with PCIe, and interconnected via 100 Gbps EDR InfiniBand with RDMA and 10 Gbps Ethernet. We use up to 4 GPU nodes in the evaluation. All used software for four models includes CUDA 11.8.0, PyTorch 2.0.0, NCCL 2.14.3, NVIDIA Apex 22.03, and Megatron-LM v3.0.2. We use data parallelism to emphasize the influence of interconnects on experiments in this section. We use FP16 precision training and set global batch size = 16 for strong scaling and micro batch size = 4 for weak scaling. The number of GPUs and batch size have such a relation: $\#GPU \times \text{micro batch size} = \text{global batch size}$.

B. Strong Scaling

1) *Evaluation on Training Time*: Figure 4(a) to 4(d) present a comprehensive analysis of the performance metrics for four different models: GPT-2-Medium, GPT2-Large, BERT-Large, and T5-Large, across various numbers of GPUs and communication protocols/interconnects. We mainly report average numbers unless otherwise stated, as models show similar performance trends.

Among all the breakdowns in training iteration time, three notable components worth discussing are the forward compute time, the backward compute time, and the backward parameters sync time (i.e., AllReduce by NCCL).

The forward compute time, representing the duration of the forward propagation step, is shown by the length of the bottom stacked bar in each subfigure. The results indicate that as the number of GPUs increases, all models' **forward compute time** diminish, achieving a **strong scaling efficiency of 56.82%**. This reduction in forward compute time signifies the increased computational power and parallel processing efficiency of utilizing multiple GPUs. Consequently, more GPUs lead to faster forward propagation time.

Similarly, the second bottom stacked bar in each subfigure represents the backward compute time, which measures the duration of the backward propagation step. As observed, the **backward compute time** diminishes with the increasing number of GPUs for all models, achieving **strong scaling efficiency of 71.71%**. This reduction in backward compute time suggests that leveraging more GPUs enables efficient parallel processing during the backward propagation step, leading to faster gradient computations and enhanced computing speed.

Observation 1: The forward and backward compute processes in LLM training can achieve a strong scaling efficiency of 56.82% and 71.71%, respectively.

However, while the forward and backward compute time diminishes with more GPUs, it is crucial to consider the trade-off associated with the backward parameter synchronization time via AllReduce, which is required to synchronize the gradients across all GPUs during the parameter update phase. As the number of GPUs increases, AllReduce time also increases significantly by 4.66x with doubled number of GPUs, mainly due to the additional communication overhead in synchronizing gradients across more GPUs. Notably, AllReduce time varies significantly depending on the communication protocol and the underlying interconnects used in the GPU cluster. Different interconnects and technologies, such as RDMA or TCP/IP, may exhibit varying latencies and bandwidths, impacting the efficiency of gradient synchronization. Consequently, AllReduce time may exhibit variability across GPU clusters and communication setups. Using 8 A100 GPUs, **AllReduce time takes up 53.4%, 82.48%, and 91.72% of iteration time for RDMA, IPoIB, and TCP/IP, respectively.**

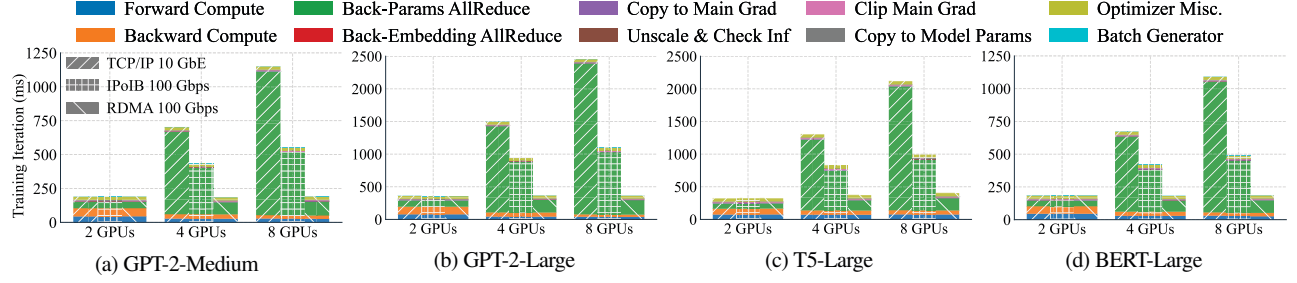


Fig. 4: Training Time Breakdown for Each Iteration under Strong Scaling.

Observation 2: AllReduce communication operation in the backward parameter synchronization step takes up most training time in each iteration, with 53.4%, 82.48%, and 91.72% for RDMA, IPoIB, and TCP/IP, respectively.

The results in Figure 4 highlight the trade-off between reduced forward and backward compute time and increased AllReduce time when scaling the models across multiple GPUs. While leveraging more GPUs leads to faster forward and backward computations, the increased communication overhead during gradient synchronization can impact overall training time. Therefore, it becomes essential to carefully consider the computation intensity required with training configurations and the choice of communication protocols and the underlying interconnects to optimize the training process. The scalability analysis among 2, 4, and 8 GPUs underscores the importance of balancing computational efficiency and communication overhead in distributed training. We hope it can provide valuable insights for researchers and practitioners in determining the optimal GPU configuration and communication setup for their training needs.

2) *Evaluation on Interconnect Utilization:* Since AllReduce for backward parameter synchronization takes the most training time, which can be significantly influenced by protocols and interconnects, we next analyze the interconnect utilization (achieved speed) during the experiments. The results are obtained through system counters. Here we only showcase the interconnect utilization with 8 GPUs in Figure 5. This choice is made because the experiment with 8 GPUs demonstrates the highest utilization and provides a clearer picture of the interconnect’s performance. The evaluation is based on RX (Receive) and TX (Transmit) speeds of training the four models with RDMA, IPoIB, and TCP/IP. Due to their similarity, the result of IPoIB can represent the performance of TCP/IP on high-speed interconnect to some extent.

Analyzing the data, we observe distinct performance ranges for each interconnect/protocol option. For the two protocols using 100 Gbps InfiniBand, **RDMA** shows the highest speeds, ranging from **30 to 60 Gbps for both RX and TX**. This indicates that RDMA can efficiently utilize the available bandwidth, offering the highest throughput among the tested interconnect/protocol options. On the other hand, **IPoIB** exhibits slightly lower speeds, with **RX and TX ranging from 17 to 20 Gbps**. IPoIB achieves notable performance albeit lower speed than RDMA, demonstrating effectiveness in the given scenario. In contrast, **TCP/IP** shows the lowest speeds among the interconnect/protocol options, with **RX and TX speeds ranging**

from 8 to 9 Gbps. The slowest performance of TCP/IP suggests that it may be limited in its ability to fully utilize the available bandwidth.

Observation 3: Interconnect utilization for training LLMs follows the trend – RDMA (30-60 Gbps) > IPoIB (17-20 Gbps) > TCP/IP (8-9 Gbps) in our experiments.

Furthermore, it is worth noting that few drops in RX and TX speeds are observed. These drops occur due to the checkpointing and validation processes performed during these models’ training. Checkpointing involves saving intermediate model states at specific intervals, while validation involves evaluating the model’s performance on a separate dataset. These operations can temporarily impact the communication and processing speed of the system, incurring periodic drops in RX and TX speeds.

We observe **GPT-2-Large** consistently achieves higher **RX and TX speeds (30.47 Gbps)** within the models tested than other models, like GPT-2-Medium (19.93 Gbps), BERT-Large (26.48 Gbps), and T5-Large (24.19 Gbps). This is because it is relatively large among all models in our experiments and therefore requires more data communication. T5-Large suffers more severe periodic utilization drops albeit in comparable size, which hinders its overall interconnect utilization. This suggests that GPT-2-Large is more efficient in utilizing the interconnect bandwidth and demonstrates better scaling performance in the given strong scaling scenario.

Observation 4: Generally, larger LLMs have higher interconnect utilization requirements. GPT-2-Large consistently achieves higher RX and TX speeds at an average bandwidth of 30.47 Gbps in our experiments.

To summarize, there leaves to be desired for the overall interconnect utilization, given we have 100 Gbps InfiniBand and 10 Gbps Ethernet. Therefore, future analyses may explore the interplay among the models, GPU configurations, and interconnect/protocol options to gain a deeper insight into their performance characteristics and identify strategies for maximizing interconnect utilization.

C. Weak Scaling

1) *Evaluation on Training Time:* In this section, we evaluate the weak scaling of those models across different GPU configurations, specifically 2, 4, and 8 GPUs. We report average numbers due to similar strong scaling trends unless stated otherwise. From Figure 6, we can see that training iteration time depends mainly on the

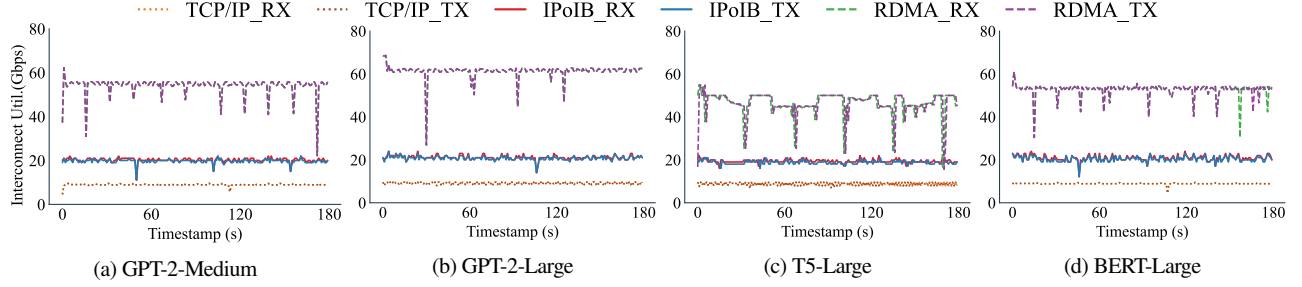


Fig. 5: Interconnect Utilization under Strong Scaling.

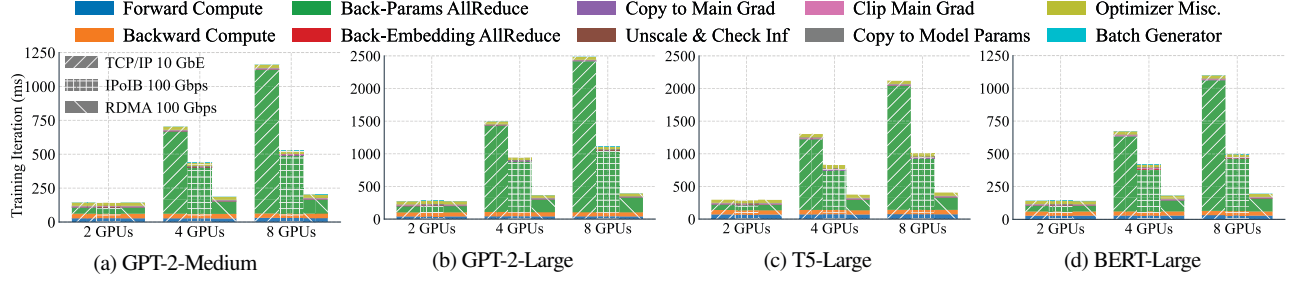


Fig. 6: Training Time Breakdown for Each Iteration under Weak Scaling.

forward compute time, backward compute time, and backward parameter synchronization time. As weak scaling refers to the ability of a parallel system to maintain a fixed problem size per GPU, we set a fixed micro batch size per GPU to four.

Under varying numbers of GPUs, all models demonstrated excellent scalability regarding **forward compute time and backward compute time**, with **97% and 99.47%** for each. As the GPU count increased from 2 to 4 and 8, these models continued to scale well, with forward and backward compute time remaining relatively constant. This indicates that all models can handle larger problem sizes without significantly increasing compute time. The ability to maintain a consistent execution time across different GPU configurations suggests pronounced weak scaling characteristics for forward and backward computation in these models.

Observation 5: Forward and backward compute time remains near consistent and can achieve 97% and 99.47% in weak scaling efficiency for distributed LLM training.

However, it is worth noting that the models showcased varying weak scaling characteristics regarding AllReduce operation. Across all models, the overall trend remains the same: AllReduce time is heavily influenced by the protocols/interconnect. Specifically, **RDMA outperforms IPoIB and TCP/IP** by 4.07x and 8.73x, respectively, leading to **2.51x and 4.79x faster training iterations**.

Observation 6: In weak scaling evaluation, AllReduce time for LLM parameter synchronization remains heavily influenced by protocols/interconnects. RDMA promotes 2.51x faster training iterations than IPoIB and the performance disparity further enlarges to 4.79x compared to TCP/IP.

Among the models, GPT-2-Large and T5-Large demonstrate longer AllReduce time across all GPU configurations, indicating potential communication challenges during the parameter synchronization step. On the other hand, GPT-2-Medium and BERT-Large showcase relatively shorter AllReduce time. Using 8 A100 GPUs, AllReduce time takes up to 50.5%, 80.78%, and 91.12% of iteration time for RDMA, IPoIB, and TCP/IP, respectively.

Observation 7: For both strong and weak scaling, network communications play an important role in LLM training. In weak scaling, AllReduce time takes up to 50.5%, 80.78%, and 91.12% of iteration time for RDMA, IPoIB, and TCP/IP.

In conclusion, while AllReduce time relies heavily on interconnect types, all models' forward and backward compute time scales well. These observations highlight the importance of considering the interconnect's characteristics when assessing the scalability of distributed training for LLMs while showcasing the models' ability to scale efficiently in forward and backward computation.

2) *Evaluation on Interconnect Utilization:* Analogously, we showcase the interconnect utilization under a weak scaling experiment with 8 GPUs in Figure 7, providing insights into the performance characteristics of different models and interconnect/protocol options. With our analysis, several pivotal findings emerge.

The weak scaling trend initially follows a similar pattern to the strong scaling experiment across all models and interconnect/protocol options. As the number of GPUs increases, the interconnect utilization also improves, indicating that adding more GPUs effectively utilizes the available interconnect resources. In the experiment with 8 GPUs, we observe the highest utilization, demonstrating the scalability potential of the system.

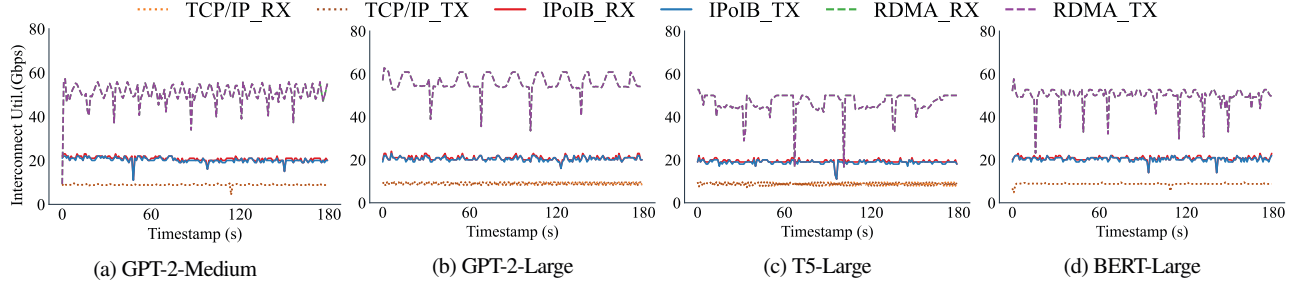


Fig. 7: Interconnect Utilization under Weak Scaling.

Among the tested models, GPT-2-Large consistently exhibits the highest utilization (28.6 Gbps), reaffirming its demand to effectively utilize the interconnect. This can be attributed to its larger model size and computational requirements, rendering more use of the available computational resources and interconnect bandwidth.

Regarding interconnect/protocol options, RDMA consistently outperforms IPoIB and TCP/IP. **RDMA**, with its efficient direct memory access capabilities, achieves the highest **RX and TX speeds (38-56 Gbps)**. **IPoIB** follows with slightly lower speeds (**17-20 Gbps**), while **TCP/IP** exhibits the lowest speeds (**8-9 Gbps**). This hierarchy emphasizes again the importance of choosing the appropriate interconnect/protocol option for achieving optimal interconnect utilization.

Observation 8: The interconnect utilization under different protocols/interconnects in weak scaling is analogous to that in strong scaling, with the interconnect utilization of 38-56 Gbps for RDMA, 17-20 Gbps for IPoIB, and 8-9 Gbps for TCP/IP.

Additionally, periodic drops in RX and TX speeds are observed in the data, analogous to the strong scaling experiment. These drops occur due to the checkpointing and loss validation processes necessary to maintain the training process's integrity and accuracy. A heavier workload of 8 GPUs incurs more pronounced RX and TX speed drops. Despite being temporary, these drops reflect the trade-off between performance and ensuring the quality of the training process.

In conclusion, the weak scaling experiment with 8 GPUs reinforces the importance of interconnect utilization in scaling scenarios. The result highlights GPT-2-Large's higher interconnect utilization, the advantage of RDMA over IPoIB and TCP/IP, and the presence of periodic drops in Recv and Send speeds. These findings contribute to a deeper understanding of the performance characteristics of models and interconnect/protocol options in the context of weak scaling. Further analysis and experimentation can build upon these insights to optimize interconnect utilization in different scaling scenarios.

D. Summary

As illustrated in Figure 8, we further evaluate these models with increased batch sizes until out-of-memory (OOM) and observe a similar trend where communication takes a large portion of the iteration time. Notably, this figure demonstrates that even though increasing the batch sizes can result in a reduced proportion of communication time in the overall iteration time (amortized by the prolonged

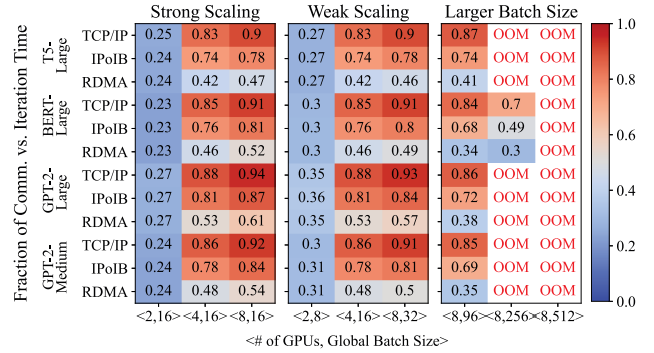


Fig. 8: Fraction of Comm. vs. Iteration Time with Larger Batch Sizes

computation time), the communication time can still occupy at least 34% of iteration time (except for BERT-Large, as it allows for a much larger batch size). This observation highlights a lower bound of the communication time proportion since it investigates until the maximum batch size a model can train with at the given scale.

V. RELATED WORK

Optimizing and characterizing big data and deep learning workloads on high-speed networks have been fruitful research topics recently. Studies on RDMA-optimized versions of Hadoop [18], [19], Spark [20], [21], TensorFlow [22], [23], and PyTorch [24] demonstrate that Big Data and Deep Learning technologies can achieve remarkable performance gains through utilizing these high-performance interconnects. DLoBD [25] investigates the impact of high-performance interconnects on integrating deep learning systems (e.g., TensorFlow and Caffe) with big data processing stacks (e.g., Hadoop and Spark) in HPC clusters. Megatron-LM [10] presents techniques for training very large transformer models with billions of parameters using an efficient intra-layer model parallel approach. Compared to these investigations, this paper focuses on the emerging LLMs training within the context of performance characterization using high-speed interconnects, which delves into a more detailed analysis and specifically addresses the challenges and opportunities associated with their integration.

Research efforts have endeavored [26], [27] to provide standardized and objective performance evaluations of machine learning systems across different hardware platforms and software frameworks. MLPerf [28], [29] is a major success in establishing a standardized framework for researchers in academia and industry

to submit state-of-the-art benchmark results. While the MLPerf benchmarking framework may not directly address the impact of high-speed interconnects on LLM training, the results and methodologies provided by this paper complement existing ML benchmarking efforts for guiding the experiments and performance evaluations of LLMs on high-speed networks.

VI. CONCLUSION AND FUTURE WORK

Overall, this paper contributes to understanding the performance characteristics of large language models over high-speed interconnects. We extensively explore and summarize the role communication plays in distributed LLM training. The results can inform the design and deployment of efficient systems to support the growing demand for LLM applications. The evaluation results enable us to yield valuable observations. These findings indicate that strong scaling and weak scaling experiments exhibit similar trends, emphasizing the influence of interconnect/protocol in distributed training and the importance of effective interconnect utilization. Some of the pivotal observations include:

- ① Forward and backward compute times scale well for both strong and weak scaling. However, distributed training with data parallelism for LLM brings scalability challenges for communication during backpropagation.
- ② Faster interconnects/protocols can significantly reduce the distributed training time for LLMs. In particular, our numbers show that GPUDirect RDMA outperforms IPOB and TCP/IP by 2.51x and 4.79x on average, regarding training performance.
- ③ LLMs with more parameters tend to show higher interconnect utilization requirements, but there is still room for the overall interconnect utilization to be improved even under heavy LLM workloads.

Some of the future work may include investigating other parallelism methods like model parallelism, exploring distributed training behavior for even larger models at larger scales, and developing techniques to further optimize interconnect utilization.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable insights, thoughtful comments, and constructive feedback. Their expertise and thorough evaluation significantly contributed to the improvement of this work. This work was supported in part by an Amazon Research Award. Part of this research was conducted using Pinnacles (NSF MRI, #2019144) at the Cyberinfrastructure and Research Technologies (CIRT) at the University of California, Merced.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805 (2018).
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, I. Polosukhin, Attention is All You Need, *Advances in neural information processing systems* 30 (2017).
- [3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language Models are Unsupervised Multitask Learners, *OpenAI blog* 1 (8) (2019) 9.
- [4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language Models are Few-shot Learners, *Advances in neural information processing systems* 33 (2020) 1877–1901.
- [5] GPT-4 Technical Report, <https://openai.com/research/gpt-4>.
- [6] A. Gilson, C. W. Safranek, T. Huang, V. Socrates, L. Chi, R. A. Taylor, D. Chartash, et al., How does CHATGPT Perform on the United States Medical Licensing Examination? The Implications of Large Language Models for Medical Education and Knowledge Assessment, *JMIR Medical Education* 9 (1) (2023) e45312.
- [7] D. Araci, Finbert: Financial Sentiment Analysis with Pre-trained Language Models, arXiv preprint arXiv:1908.10063 (2019).
- [8] M. G. Sousa, K. Sakiyama, L. de Souza Rodrigues, P. H. Moraes, E. R. Fernandes, E. T. Matsubara, BERT for Stock Market Sentiment Analysis, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2019, pp. 1597–1601.
- [9] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. Patwary, M. Ali, Y. Yang, Y. Zhou, Deep Learning Scaling is Predictable, Empirically, arXiv preprint arXiv:1712.00409 (2017).
- [10] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, et al., 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*, 2021, pp. 1–15.
- [11] MTIA: META'S First Generation of AI Accelerators, <https://atscaleconference.com/videos/mtia-metas-first-generation-of-ai-accelerators/>.
- [12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-To-Text Transformer, *The Journal of Machine Learning Research* 21 (1) (2020) 5485–5551.
- [13] English Wikipedia Dump, <https://dumps.wikimedia.org/enwiki/20230501/>.
- [14] V. Cerf, R. Kahn, A Protocol for Packet Network Intercommunication, *IEEE Transactions on communications* 22 (5) (1974) 637–648.
- [15] V. Kashyap, IP over InfiniBand (IPOB) Architecture, Tech. rep. (2006).
- [16] R. Recio, B. Metzler, P. Culley, J. Hilland, D. Garcia, A Remote Direct Memory Access Protocol Specification, Tech. rep. (2007).
- [17] MRI: Acquisition of Pinnacles – Raising Research Computing to New Heights in California's Central Valley, https://www.nsf.gov/awardsearch/showAward?AWD_ID=2019144&HistoricalAwards=false.
- [18] X. Lu, N. S. Islam, M. W. Rahman, J. Jose, H. Subramoni, H. Wang, D. K. Panda, High-Performance Design of Hadoop RPC with RDMA over InfiniBand, in: *The Proceedings of IEEE 42nd International Conference on Parallel Processing (ICPP)*, France, 2013.
- [19] N. S. Islam, X. Lu, M. W. Rahman, D. K. Panda, Can Parallel Replication Benefit Hadoop Distributed File System for High Performance Interconnects?, in: *The Proceedings of IEEE 21st Annual Symposium on High-Performance Interconnects (HOTI)*, San Jose, CA, 2013. doi:10.1109/HOTI.2013.24.
- [20] X. Lu, M. W. U. Rahman, N. Islam, D. Shankar, D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, in: *2014 IEEE 22nd Annual Symposium on High-Performance Interconnects*, 2014, pp. 9–16.
- [21] X. Lu, D. Shankar, S. Gugnani, D. K. D. K. Panda, High-Performance Design of Apache Spark with RDMA and Its Benefits on Various Workloads, in: *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 253–262.
- [22] R. Biswas, X. Lu, D. K. Panda, Accelerating TensorFlow with Adaptive RDMA-Based gRPC, in: *Proceedings of IEEE International Conference on High Performance Computing, HiPC '18*, IEEE, 2018, pp. 2–11.
- [23] A. Sergeev, M. Del Balso, Horovod: Fast and Easy Distributed Deep Learning in TensorFlow, arXiv preprint arXiv:1802.05799 (2018).
- [24] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, et al., Pytorch Distributed: Experiences on Accelerating Data Parallel Training, arXiv preprint arXiv:2006.15704 (2020).
- [25] X. Lu, H. Shi, M. H. Javed, R. Biswas, D. K. Panda, Characterizing Deep Learning over Big Data (DLBD) Stacks on RDMA-Capable Networks, in: *Proceedings of Annual Symposium on High-Performance Interconnects, HotI '17*, IEEE Computer Society, 2017, pp. 87–94.
- [26] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, J. Leskovec, Open Graph Benchmark: Datasets for Machine Learning on Graphs, *Advances in neural information processing systems* 33 (2020) 22118–22133.
- [27] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, Deep Learning for Precipitation Nowcasting: A Benchmark and a New Model, *Advances in neural information processing systems* 30 (2017).
- [28] 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, *Proceedings of Machine Learning and Systems* 2 (2020) 336–349.
- [29] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, et al., MLPerf Inference Benchmark, in: *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, IEEE, 2020, pp. 446–459.