Performance Characterization of Large Language Models on High-Speed Interconnects

Hot Interconnects 2023

Hao Qi*, Liuyao Dai*, Weicong Chen*, Zhen Jia+, and Xiaoyi Lu*

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

zhej@amazon.com

http://padsys.org/

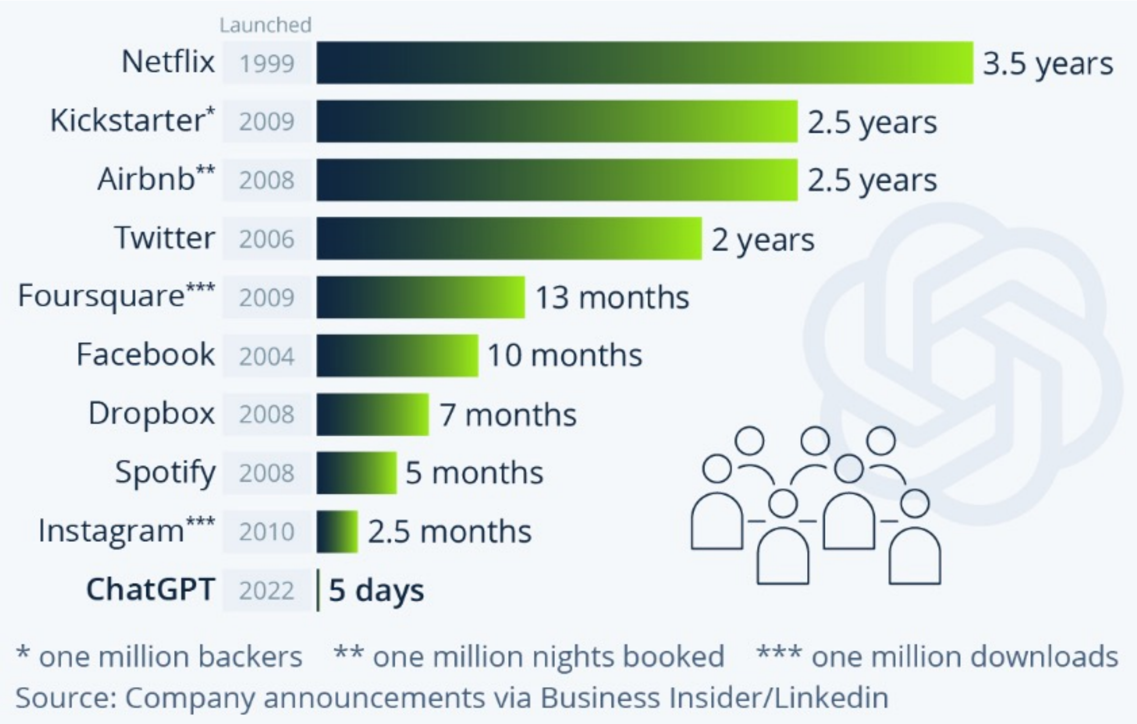
*Department of Computer Science and Engineering, University of California, Merced
+Amazon Web Service, Santa Clara

Agenda

- Introduction and Background
- Characterization Methodology
- Evaluation Results
- Conclusion and Future Work

The Rise of Large Language Models (LLMs)

- Large Language Models (LLMs) generate human-like text and perform various NLP tasks
- Transformer-based models like **GPT, BERT, and T5** have revolutionized natural language processing
- Applications: language translation, text generation, sentiment analysis, etc
- Challenges: **millions to trillions of parameters**, requiring substantial computational power and memory

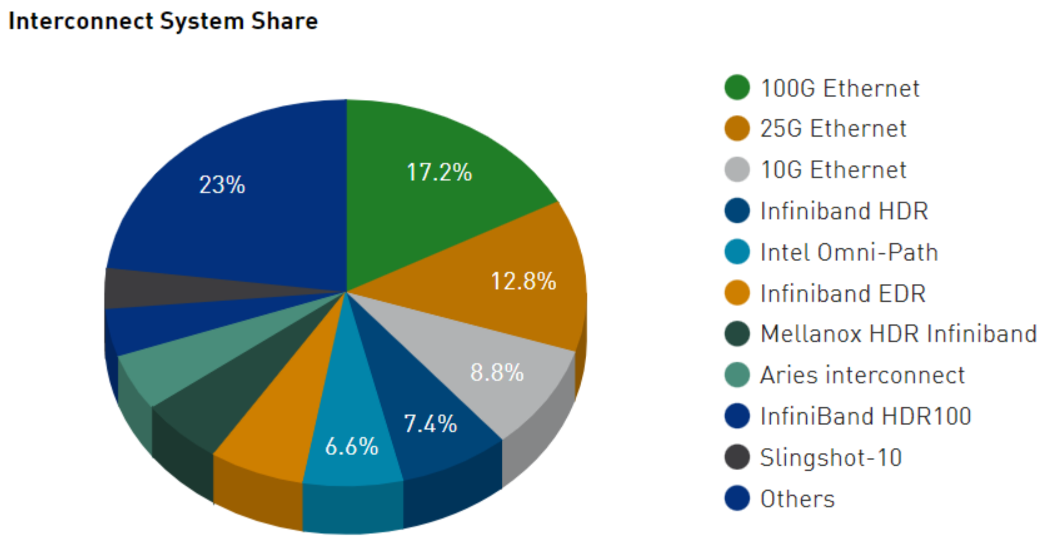


ChatGPT only takes 5 days to reach 1M users

https://www.digitalinformationworld.com/2023/01/chat-gpt-achieved-one-million-users-in.html

High-Speed Interconnects

- Types: **Ethernet, InfiniBand**, Omni-Path, etc
- Function: facilitating **communication** among GPUs and nodes; reducing communication latency
- Impact on LLM Training: performance enhancement, scalability, efficiency
- Challenges: optimization, utilization, compatibility with various Models

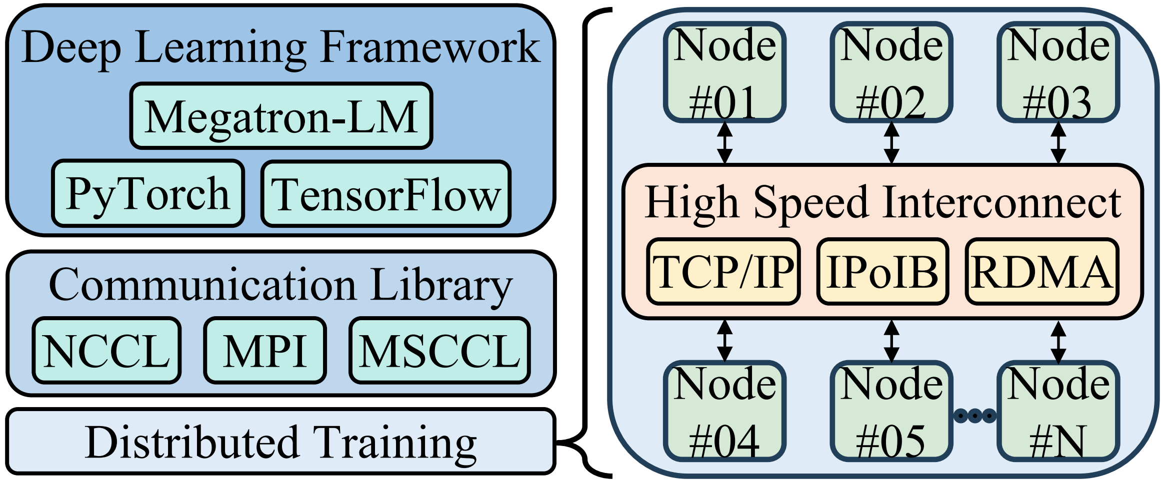


Interconnect System Share as of June 2023

https://www.top500.org/statistics/list/

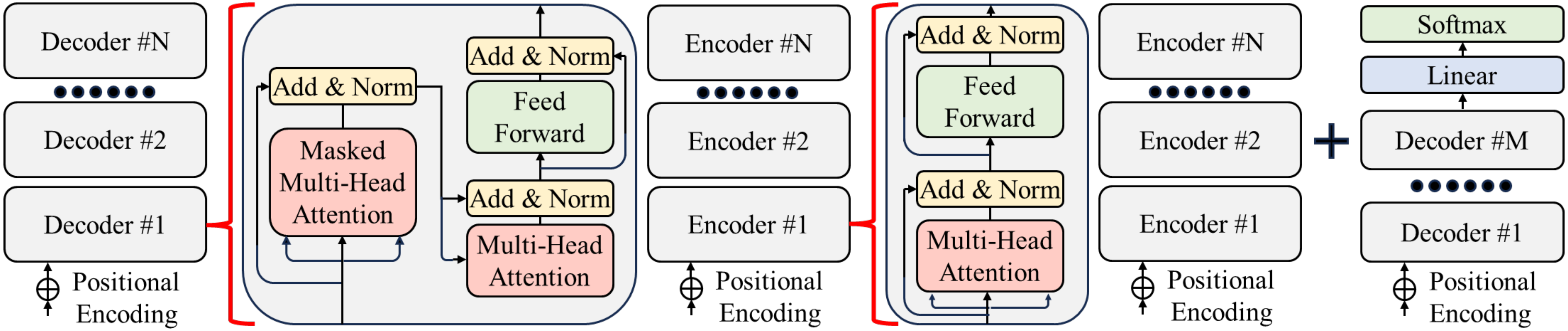
Distributed Training and the Role of High-Speed Interconnects

- **Distributed training** partitions models and data, allowing parallel training
- High-speed interconnects facilitate efficient data transfer
- Communication and coordination for fast and scalable communication
- The paper's focus: LLMs’ training performance over different high-speed **interconnects** and communication **protocols**



Communication and Interconnect in Distributed Training

Overview of Selected Models



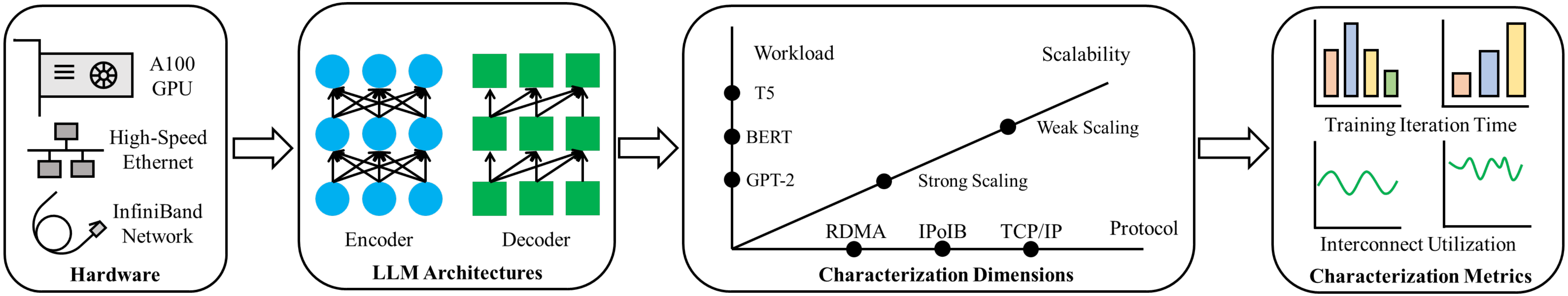
GPT: Decoder Architecture BERT: Encoder Architecture T5: Encoder & Decoder Architecture

- **GPT (Generative Pre-trained Transformer):** Decoder architecture, used for text generation, summarization, and text completion. Example variants include GPT-2, GPT-3, and ChatGPT
- **BERT (Bidirectional Encoder Representations from Transformers):** Encoder architecture, excels in text classification, named entity recognition, and sentiment analysis. Example variants include BERT-Base and BERT-Large
- **T5 (Text-To-Text Transfer Transformer):** Encoder & Decoder architecture, known for text-to-text transfer learning, used for machine translation, question-answering, and document classification

Agenda

- Introduction and Background
- Characterization Methodology
- Evaluation Results
- Conclusion and Future Work

Methodology Overview for Characterization



Characterization Methodology

- **Characterization Dimensions: Workload, scalability, and interconnects/protocols**
- **LLM Architectures:** GPT, BERT, and T5 models
- **Training Scalability:** Evaluation of strong and weak scaling aspects
- **Interconnect Technologies:** Exploration of RDMA, IPoIB, and TCP/IP
- **Communication Latency & Bandwidth Utilization:** Key metrics to quantify benefits and limitations of each interconnect/protocol option

Methodology Overview for Characterization

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

Detailed Comparison of Selected LLMs

- **Models Evaluated:** GPT-2-Medium, GPT-2-Large, BERT-Large, T5-Large
- **Framework:** We leverage the Megatron-LM as our primary distributed training framework. It provides efficient and scalable implementations of distributed training algorithms, making it an ideal choice for our investigation
- **Dataset:** We utilize the enwiki 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

Agenda

- Introduction and Background
- Characterization Methodology
- Evaluation Results
- Conclusion and Future Work

Experimental Setup for Evaluation

Cluster Configuration: NSF-funded Pinnacles cluster at UC Merced, 8 GPU nodes.

Node Specifications:

- Two Intel 28-Core Xeon Gold 6330 CPUs (2.0GHz)
- 256GB DRAM
- 2x NVIDIA Tesla A100 40GB GPUs with PCIe
- Interconnected via 100Gbps EDR InfiniBand with RDMA and 10Gbps Ethernet

Evaluation Scale: Up to 4 GPU nodes used in the evaluation

Software: CUDA 11.8.0, PyTorch 2.0.0, NCCL 2.14.3, NVIDIA Apex 22.03, and Megatron-LM v3.0.2

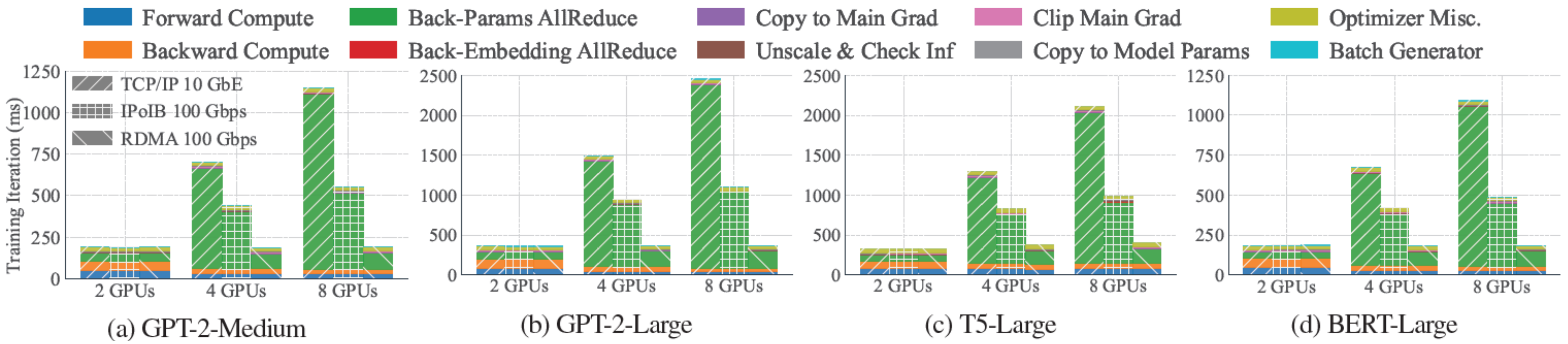
Training Approach: Data parallelism with FP16 precision training

Batch Size Configuration:

- Strong Scaling: Global batch size = 16
- Weak Scaling: Micro batch size = 4
- Relation: #GPU × micro batch size = global batch size

https://ucmerced.github.io/hpc_docs/#/Pinnacles

Training Time Breakdown under Strong Scaling

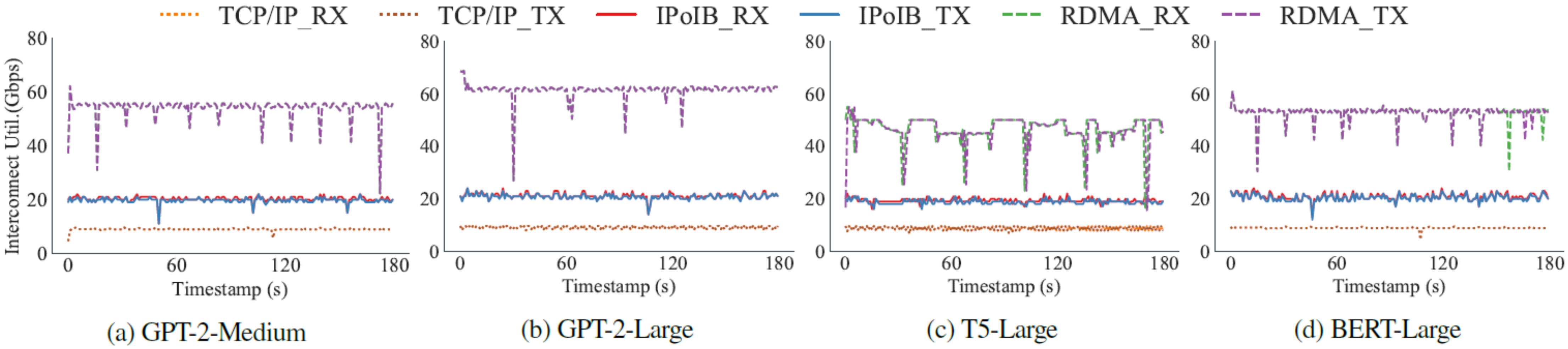


Training Time Breakdown for Each Iteration under Strong Scaling.

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

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.

Interconnect Utilization under Strong Scaling

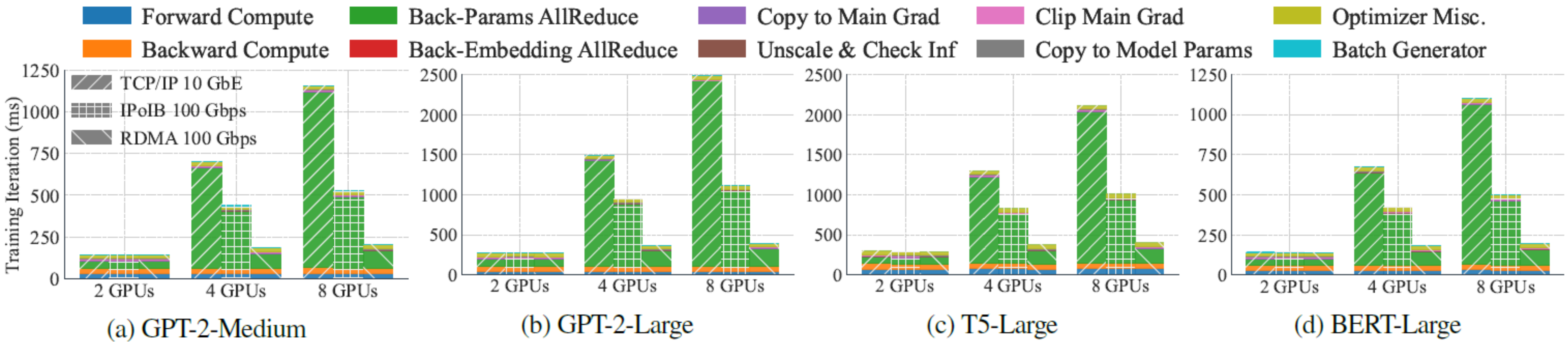


Interconnect Utilization under Strong Scaling.

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

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 30.47Gbps in our experiments.

Training Time Breakdown under Weak Scaling



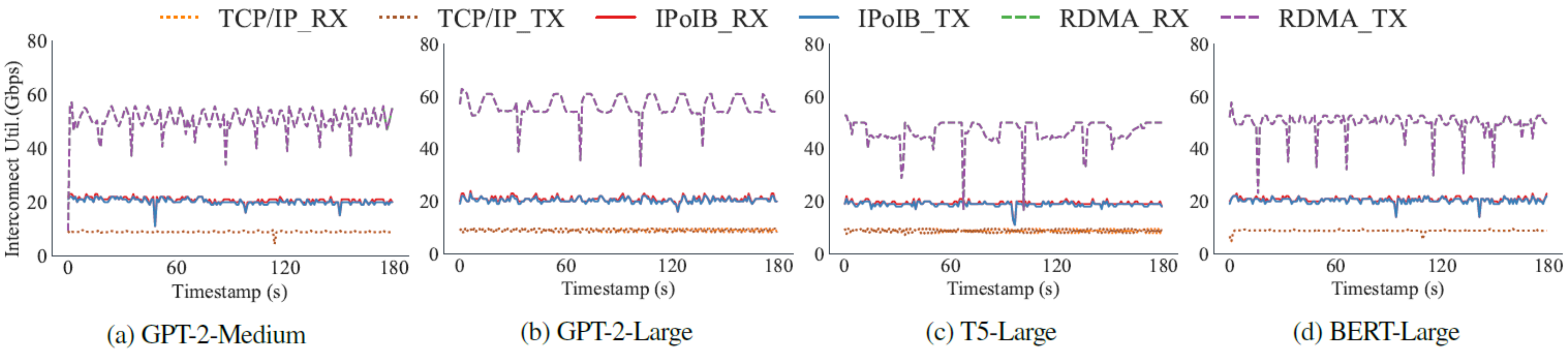
Training Time Breakdown for Each Iteration under Weak Scaling.

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

Observation 6: In weak scaling evaluation, AllReduce time for LLMparameter 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.

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.

Interconnect Utilization under Weak Scaling

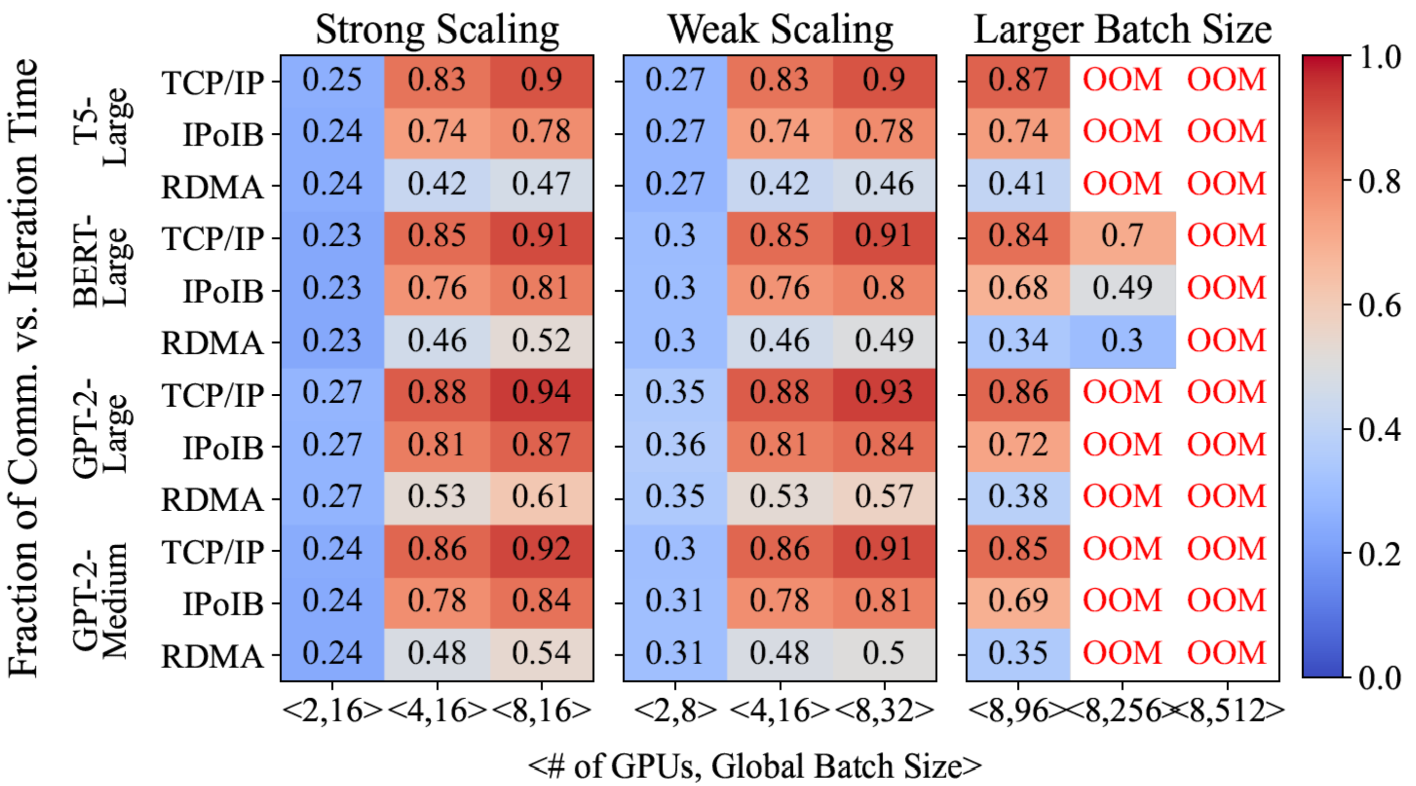


Interconnect Utilization under Weak Scaling.

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

Communication Time with Larger Batch Sizes

- Communication takes a significant portion of the iteration time, even with increased batch sizes.
- Communication time proportion can still occupy at least 34% of iteration time except for BERT-Large.
- Increasing batch sizes reduces the proportion of communication time in overall iteration time. The heatmap shows a lower bound.
- Communication time remains a critical factor to consider, even when optimizing batch sizes for training efficiency.



Fraction of Comm. vs. Iteration Time with Larger Batch Sizes.

Agenda

- Introduction and Background
- Characterization Methodology
- Evaluation Results
- Conclusion and Future Work

Conclusion and Future Work

Key Contributions:

- Exploration of communication's role in distributed LLM training.
- The results can inform design and deployment of efficient systems for LLMs.

Pivotal Observations:

- Strong and weak scaling exhibit similar trends; interconnect/protocol influence is crucial.
- Forward and backward compute times scale well, but scalability challenges exist for communication during backpropagation.
- Faster interconnects/protocols (e.g., GPUDirect RDMA) significantly reduce training time.
- LLMs with more parameters show higher interconnect utilization requirements; room for improvement exists.

Future Work:

- Investigate other parallelism methods like model parallelism.
- Explore distributed training behavior for larger models at larger scales.
- Develop techniques to further optimize interconnect utilization.

Thank you!

http://padsys.org/