SimAI: Unifying Architecture Design and Performance Tuning for Large-Scale Large Language Model Training with Scalability and Precision

NSDI'25

Xizheng Wang, Qingxu Li, Yichi Xu, Heyang Zhou and Gang Lu, Alibaba Cloud;
Dan Li, Tsinghua University; Li Chen, Zhongguancun Laboratory;
Linkang Zheng, Alibaba Cloud and South China University of Technology;

中关村实验室
ZGC Lab

華南理工大學
South China University of Technology

Presenter: Botai Sun

2025.5.30

Author

- **研究方向**
  - 计算机网络与系统，包括数据中心网络、网络智能和可信互联网
- **近期论文**
  - **大规模分布式系统仿真与性能优化**: Accelerating Design Space Exploration for LLM Training Systems with Multi-experiment Parallel Simulation (NSDI'25) 、SimAI (NSDI'25)
  - **网络流量分析**: Resolving Packets from Counters: Enabling Multi-scale Network Traffic Super Resolution via Composable Large Traffic Model (NSDI'25)
  - **网络自动化管理**: CEGS: Configuration Example Generalizing Synthesizer (NSDI'25)
  - **网络测量与服务发现**: SAIP: Accurate Detection of Anycast Servers with the Rise of Regional Anycast (INFOCOM'25)

Dan Li (李丹)
IEEE Fellow
Professor, Tsinghua University

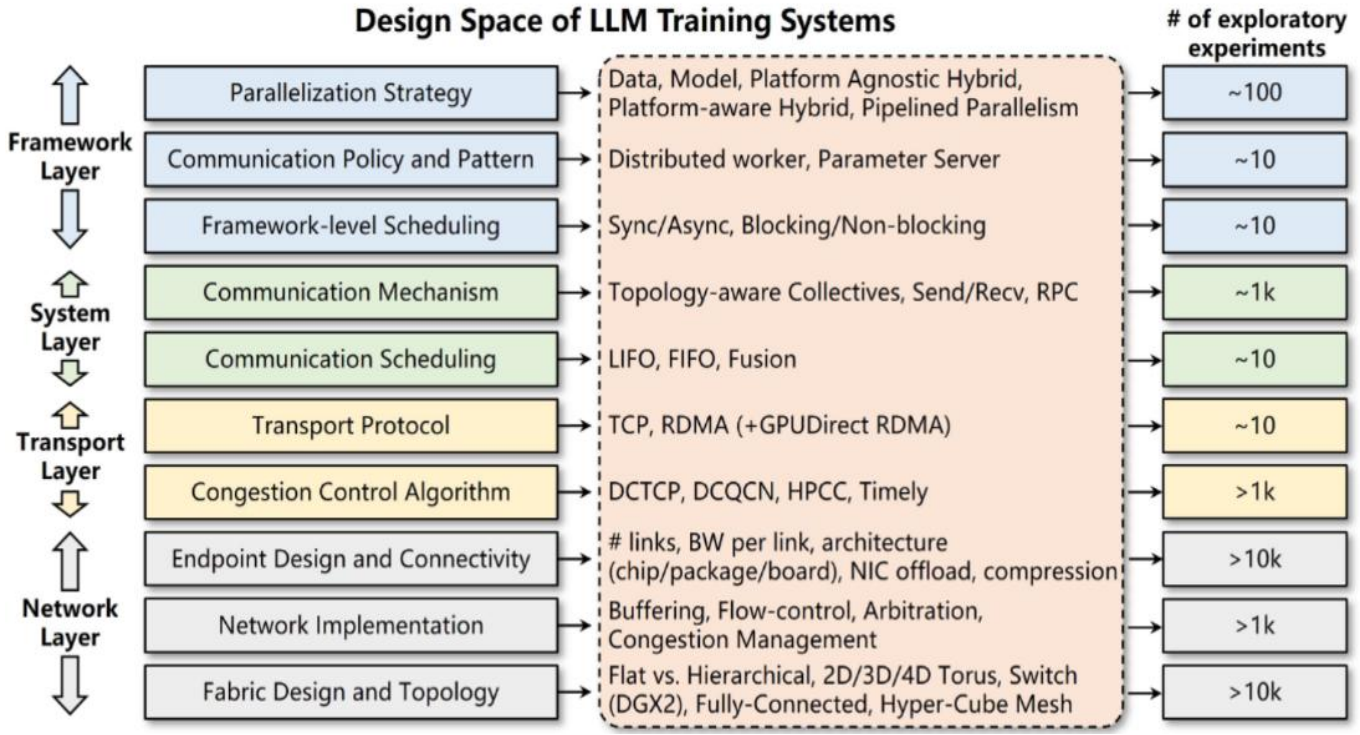
Content

- Background
- Motivation
- Related work
- Design
- Evaluation
- Conclusion

Background

需求

- LLM训练集群规模越来越大
- 框架迭代、并行训练算法迭代速度快
- 底层算力规格&网络架构多种多样
- 网络协议、拥塞控制、路由算法层出不穷

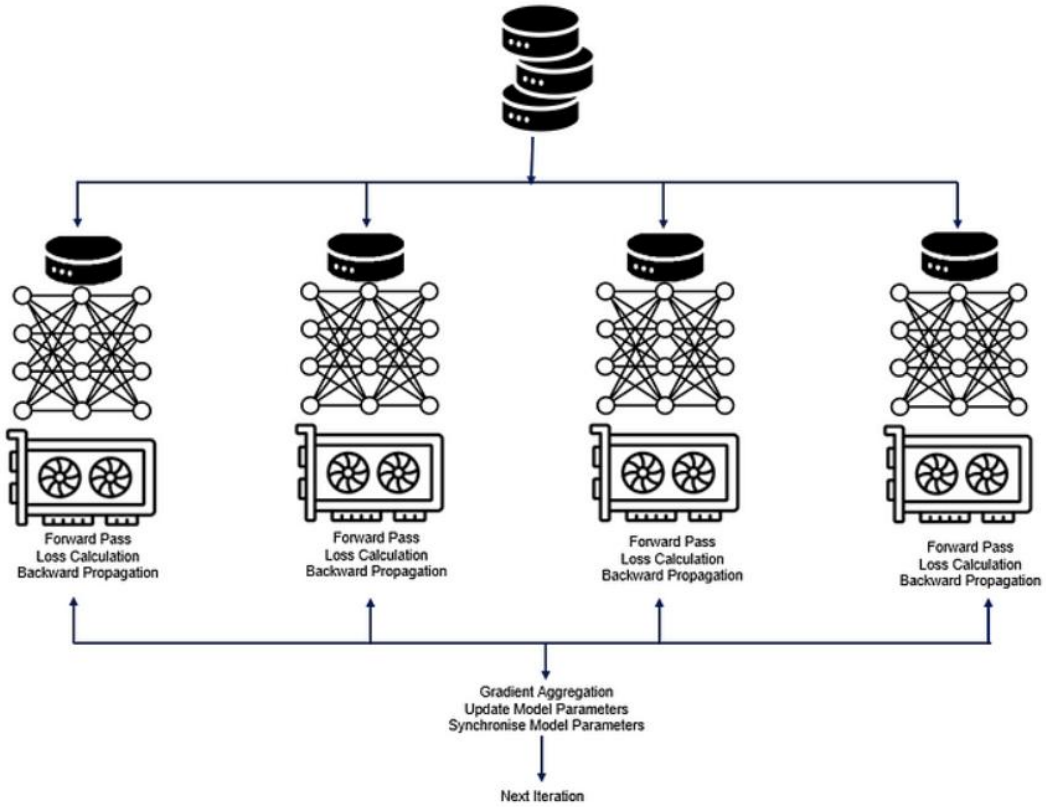


核心问题：GPU资源宝贵！

- 在物理集群上验证上述**新设计、调优和优化**困难且昂贵。

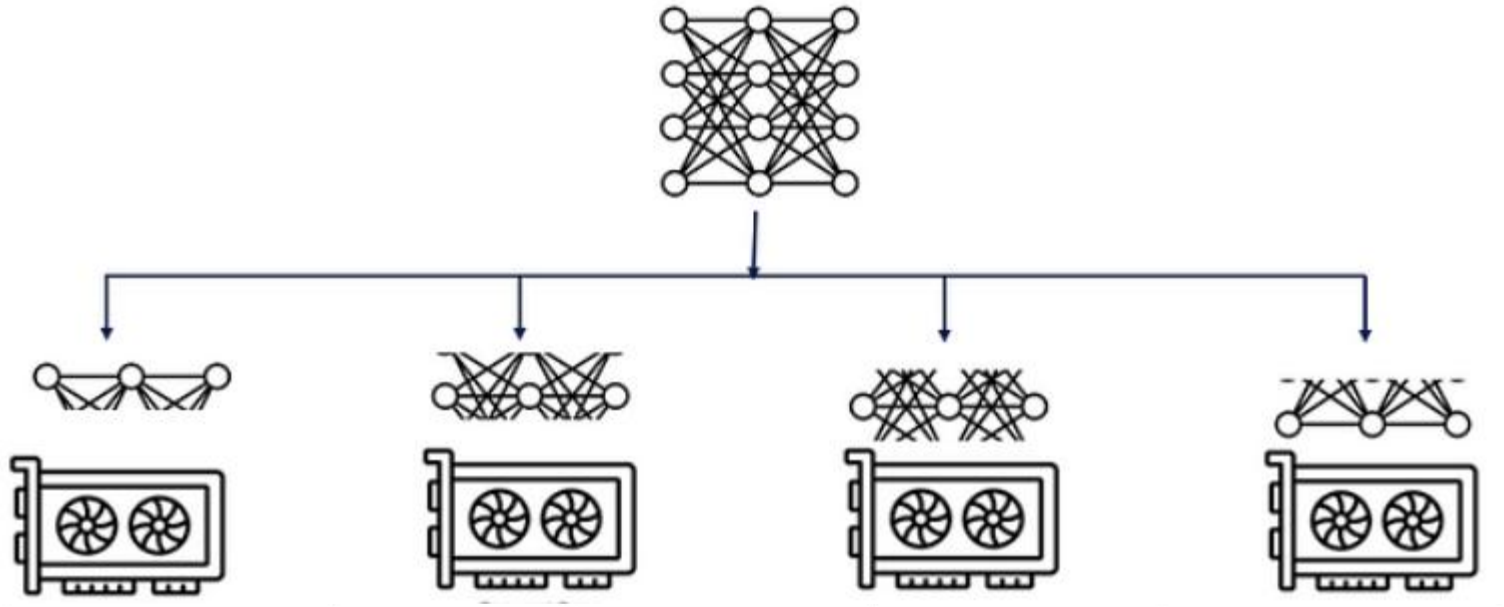
Background--Parallel strategy

- **数据并行（Data Parallelism）：**
  - 定义：每个GPU处理不同的数据批次(batch)，但模型参数相同。
  - 主要通信：反向传播时的All-Reduce操作。每个GPU需要同步其计算的梯度。



Background--Parallel strategy

- **张量并行（Tensor Parallelism）：**
  - 定义：模型参数层内分割到不同的GPU上，每个GPU处理相同的输入数据。



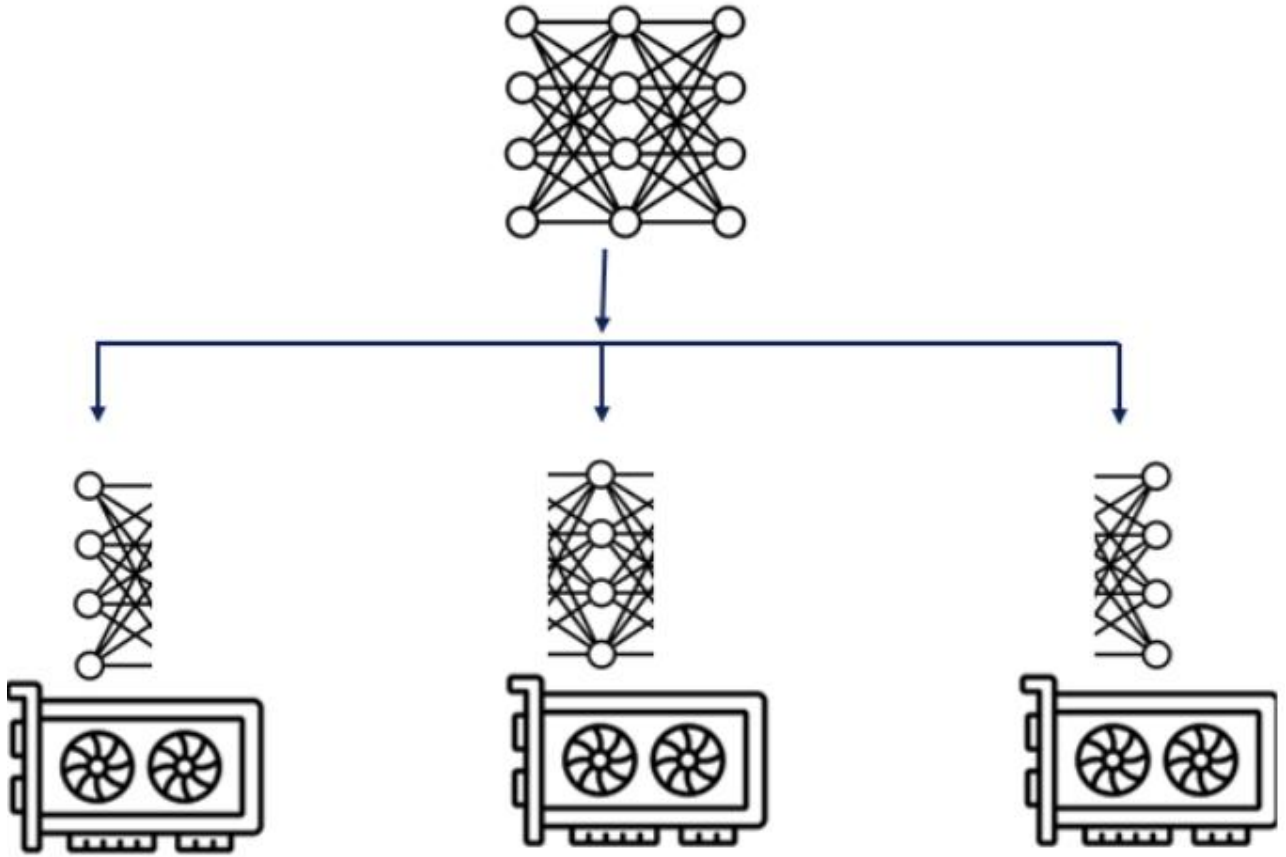
通信操作	通信数据	发生时间	通信次数	每次通信数据量
AllGather	激活值和梯度	列+列并行级联中间	每个Epoch通信 [前/反向传播次数*(相应并行级联层数-1)] 次	当前层激励（或其梯度）的数量
ReduceScatter		行+行并行级联中间		
AllReduce		行+列并行级联中间		

Background--Parallel strategy

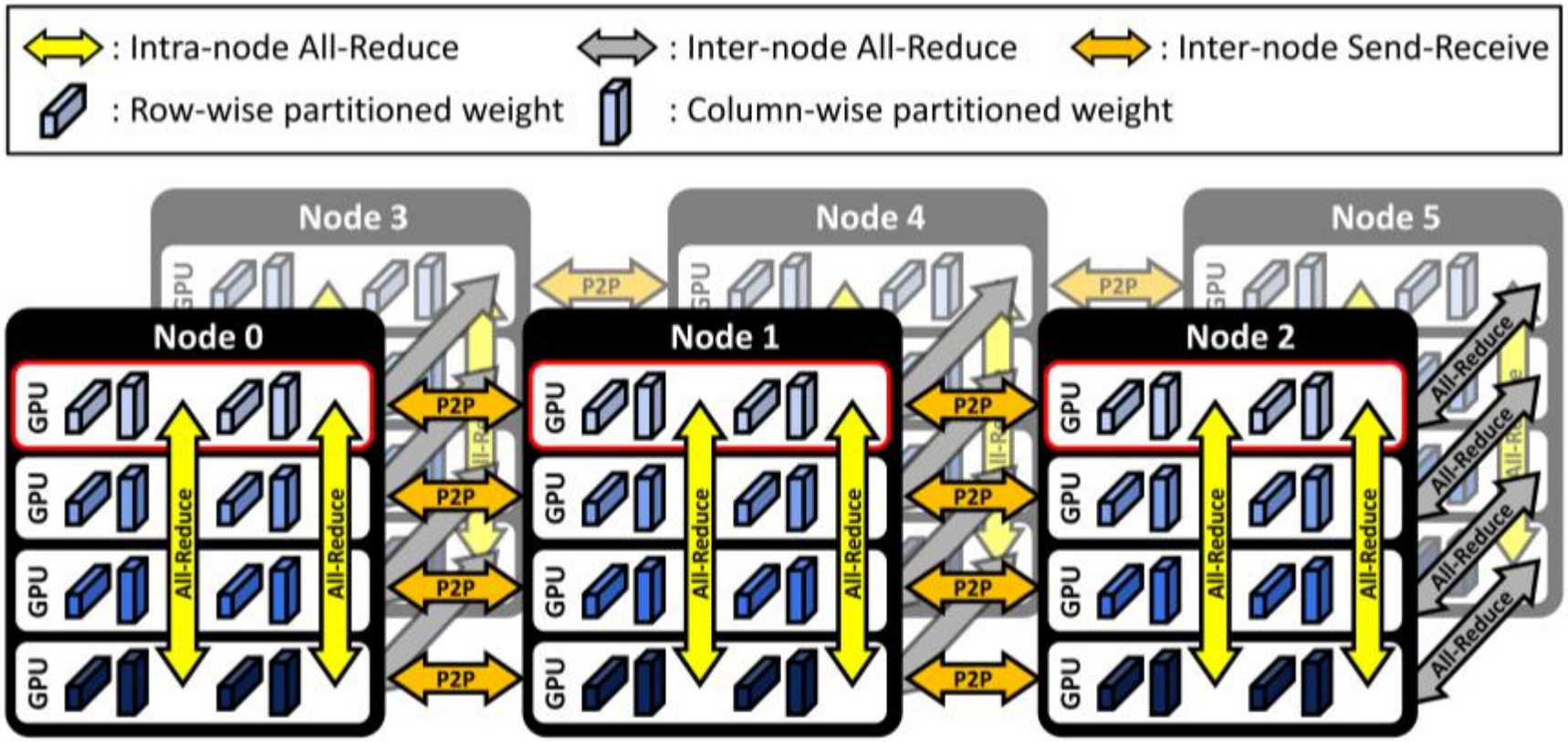
口 流水线并行（Pipeline Parallelism）：

- 定义：将模型参数按层分割，每个工作节点负责一部分连续层的前向和后向传播。

- 通信：流水线阶段的Send-Receive操作。每个GPU只需要与其相邻的GPU交换数据。



Background--Parallel strategy

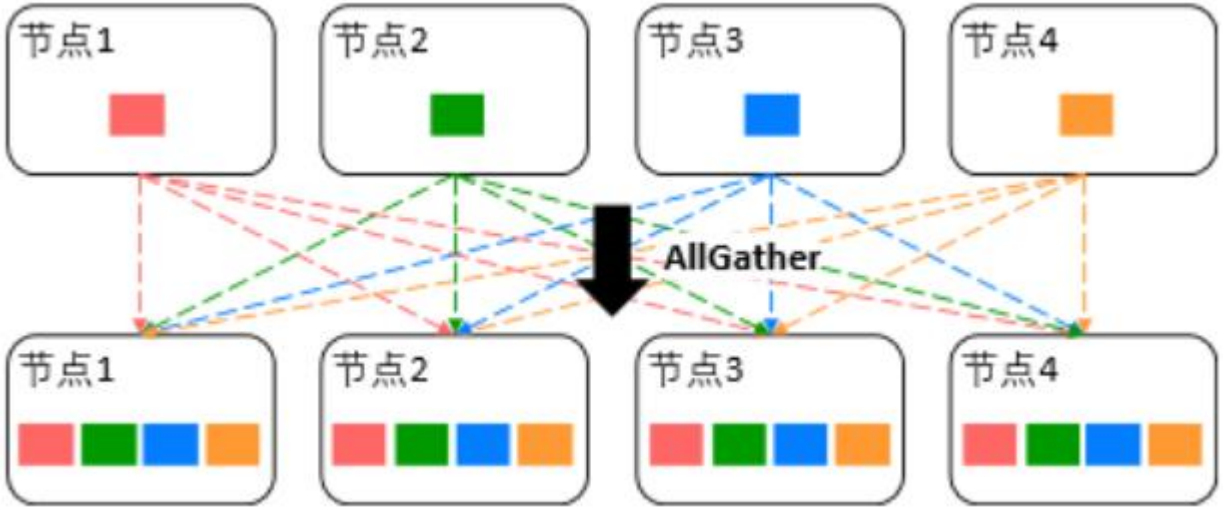


如何选择最佳的并行策略?

4路张量并行（节点内黄色All-Reduce）、2路数据并行（灰色节点间All-Reduce）和3路流水线并行性（三个节点[0,1,2]和[3,4,5]调用橙色节点间Send-Receive）。

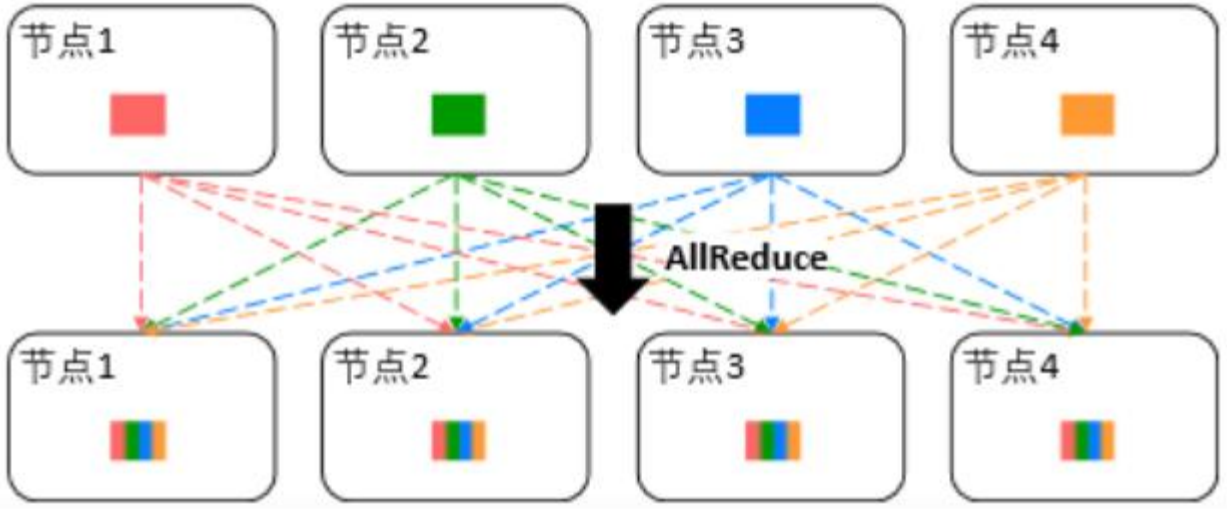
Background--Collective Communications

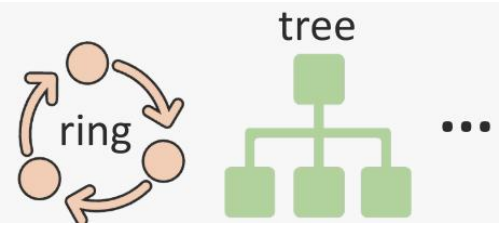
- AllGather：将每个节点上各自的数据发送到所有节点上（eg.参数共享）



如何高效的实现集合通信操作?

- AllReduce：将每个节点上的数据发送到所有节点上并进行数据规约操作（eg.梯度同步）





Background

现有模拟器局限性

- 仅关注训练的特定方面（网络or粗粒度计算）：缺乏统一视角导致在仿真复杂的端到端 LLM 训练场景时精度不足。
- 粒度问题：容量规划通常在流或作业层面进行，而性能调优通常在包级别模拟。
- 难以跟上快速发展的 LLM 架构、训练框架和硬件的步伐。

Related work

- **AI infra模拟**：ASTRA-sim(ISPASS, 2020, 2023)模拟分布式训练系统的软件和硬件协同设计堆栈；Dally通过整合模拟现代网络硬件扩展ASTRA-sim。DONS(SIGGCOM, 2023)采用面向数据的设计来提高缓存命中，但不支持RDMA传输。
- **计算模拟**：Roofline模型(Communications of the ACM, 2009)为在GPU上运行的计算内核或应用程序提供粗略的性能估计；GPGPU-Sim(ISPASS, 2009)，Accel-Sim(ISCA, 2020)供指令级模拟，但大规模部署耗时。
- **网络模拟:**
  - 数学模型估计和基于学习的近似 (如DeepQueueNet(SIGGCOM, 2022)、MimicNet(SIGGCOM, 2021))。
  - 离散事件仿真（DES）提供包级仿真，例如NS-3, OMNET++。
  - 并行离散事件模拟（PDES）用于加速（通常配置复杂，性能较差）。
  - UNISON(Eurosys, 2024)：对NS-3进行多线程改造，实现细粒度分区和负载自适应调度。
- **AI基准测试**：MLPerf, ML-Bench, AIBench (综合性); ParaDnn, DeepBench, GNNMark (特定领域)。许多基准测试在 LLM 基准测试方面存在局限性。
- **基于学习的模拟器**：通常用作端到端性能估计器 (EPE)，需要大量训练数据。SimAI (基于 DES) 可以为训练这些基于学习的模型生成有价值的数据。

Motivation

集群架构选型与论证

- GPU选型
- 机内网络架构
- 机间网络架构

AI infra技术研发测试

- 框架&模型参数
- 通信库
- 拥塞控制
- 路由算法

需要模拟一个大规模训练集群！

Motivation

模拟器的目标：

- 目标1：生成反映真实训练的**工作负载**。
- 目标2：高保真**通信仿真**。
- 目标3：高保真**计算仿真**。
- 目标4：**模拟速度快**。

SimAI: 一个**端到端**的统一模拟器，旨在**精确且高效**地仿真大规模LLM训练过程。

Design--Overview

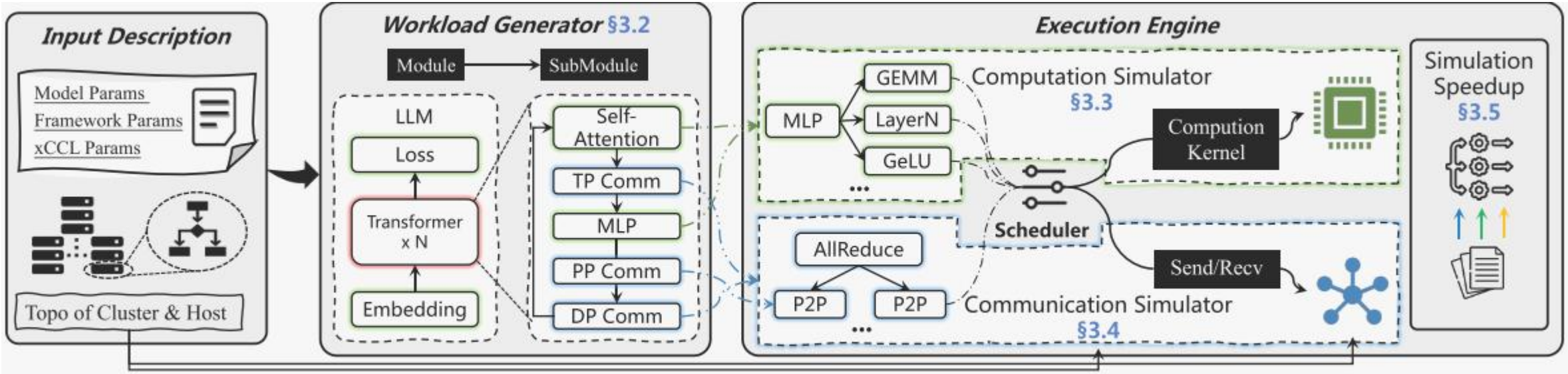


Figure 1: The SimAI architecture

- **工作负载生成器**（**W**orkload **G**enerator，SimAI-WG）：生成能代表真实 LLM 训练的精确工作负载文件。
- **执行引擎**：模拟生成的工作负载。
  - **计算仿真器**（**C**om**p**utation Simulator，SimAI-CP）： 提供精确的计算模拟。
  - **通信仿真器**（**C**om**m**unication Simulator，SimAI-CM）：提供准确的通信模拟结果。
- **仿真加速**：多线程加速模拟过程。

Design--Workload Generator

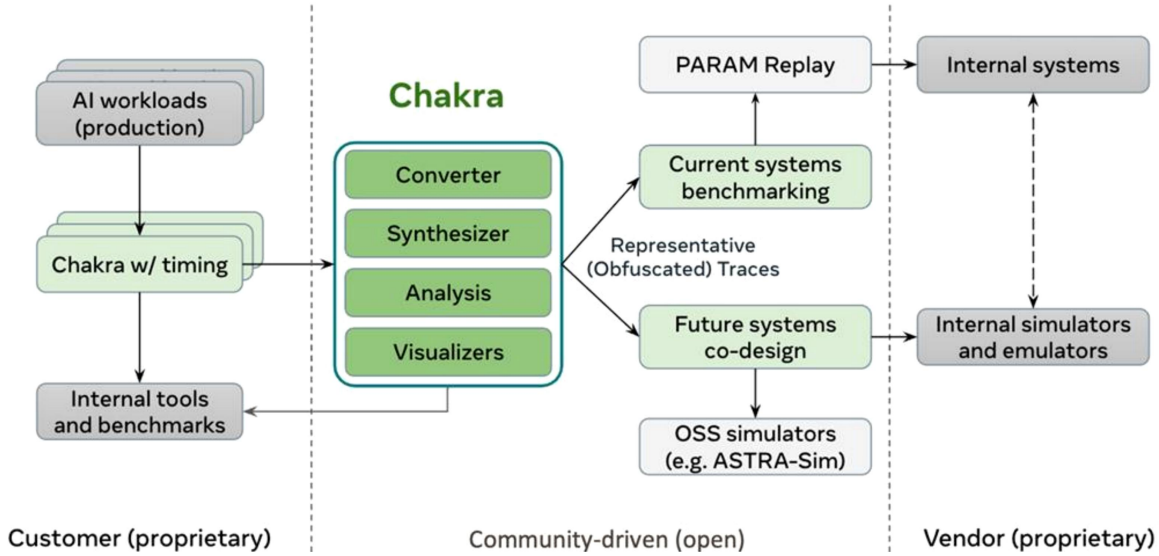
基于跟踪的方法？如Chakra

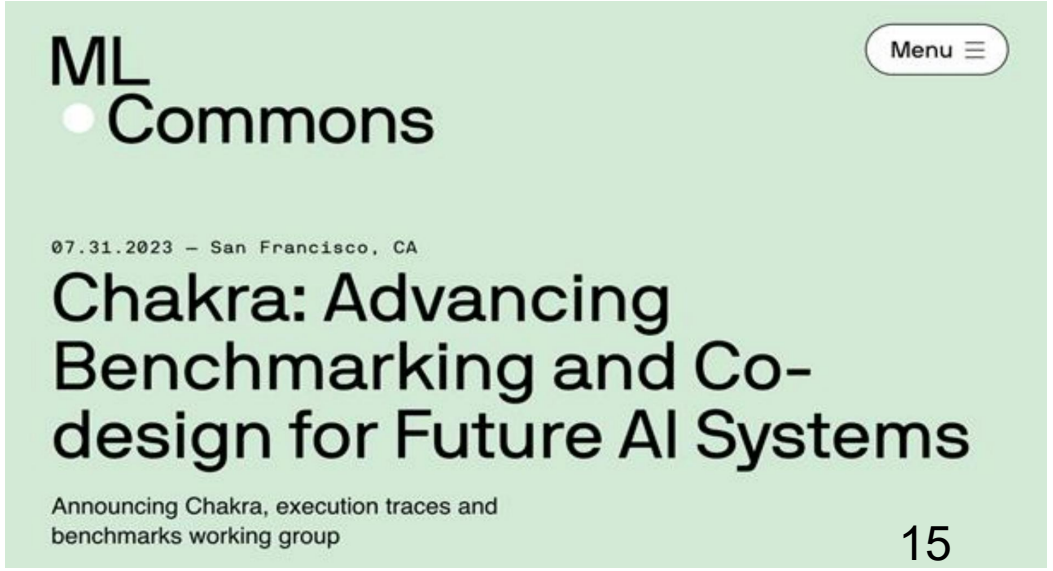
□定义：实际运行训练过程，记录其在执行过程中的一系列关键事件、操作序列或函数调用。

□问题：

- 虽然非常精确，但是需要物理集群实际运行，成本大。
- 对于模拟新的模型或配置则能力有限。

https://arxiv.org/abs/2305.14516





Design--Workload Generator

- **“劫持”现有的训练框架**：生成与真实任务一致的工作负载（在 Megatron，DeepSpeed上实现）
- **单台模拟多机思想，**对训练框架进行了修改
  - 模拟集群环境，让框架以为它在实际集群中。
  - 跳过所有多机交互逻辑。
  - 跳过所有实际通信。

	Submodules
Algorithm submodules & kernels	Grad_gather
	Embedding
	Attention_forward
	Attention_backward
	Mlp_forward
	Mlp_backward
	Layernorm_post
	Logits_parallel
	Grad_param
Collective & peer-to-peer communica-tion operations	Allreduce
	Allgather
	Reducescatter
	Send
	Recv

训练框架生成**计算和通信操作序列**，包括算法子模块和集合通信或点对点通信。

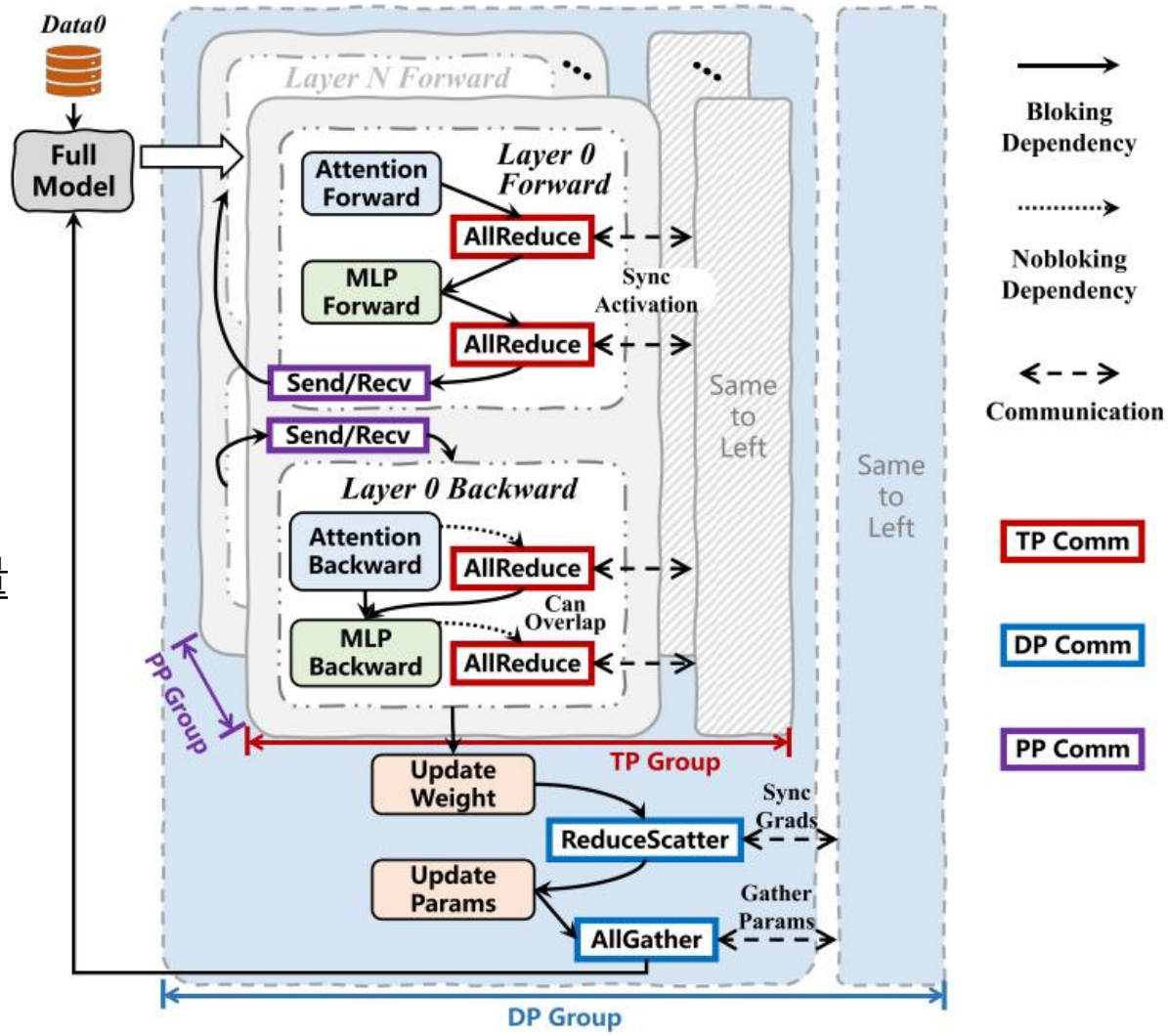
Design--Workload Generator

定义操作依赖关系

- 在工作负载文件中嵌入了依赖信息
- 依赖关系根据所使用的并行化框架确定

LLM中的通信

- 通信时间主要由各种并行技术引入，元数据交换和屏障操作可以忽略不计。
- TP、PP和EP通常具有固定的通信模式和容量不受集群大小的影响。
- DP的通信范围随着集群大小的增加而扩展（通常涉及跨数百到数千个节点的千兆级集体通信）



Design--Computation Simulation

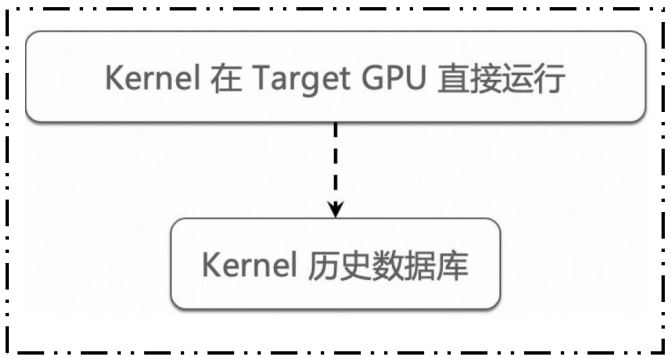
口模拟现有GPU

- 从框架中获取详细的子模块工作流。
- 维护一个操作数据库**记录子模块在相应GPU上的执行时间。**

	Submodules	Kernels
Algorithm submodules & kernels	Grad_gather	Grad_gather
	Embedding	Embedding★
	Attention_forward	Layernorm★
		Attention_QKV★
		Core_attention△
		Attention_linear★
	Attention_backward	Attention_backward
	Mlp_forward	Layernorm★
		Mlp_linear_4h△

口细粒度核心模拟

- 应对新的并行策略或优化（可能对核心进行重组或细化）。
- 设计了一个子模块-核心转换器将子模块分解为更小的核心，并对核心进行测试和记录。



GPUs_ConfigParams \ Operations	Embedding	LayerNorm	...	MLP-Linear
A100_Params1	3425(μs)	715(μs)	...	...
A100_Params2	3157(μs)	695(μs)	...	...
...	...	...	...	...

Table 4: SimAI-CP database format

Design--Computation Simulation

口模拟未发布或无法直接物理测试的GPU（通过SimAI-CP-Model）

- 云服务提供商评估新的GPU是否值得投资。

口方法

- 利用硬件规格：目标未发布GPU的核心规格（如理论浮点运算能力FLOPS、内存带宽）以及一个已知架构相似的现有GPU（作为基准）的规格。
- 区分核心类型：不同的内核有不同的性能瓶颈，通常分为两类：**计算密集型或内存带宽密集型。**
- 不同预测公式：乘以一个系数来进行缩放

$$\text{Time}_{\text{Comp\_New}} = \frac{\text{FLOPS}_{\text{Comp\_New}}}{\text{FLOPS}_{\text{Comp\_Known}}} \times \text{Time}_{\text{Comp\_Known}}$$

$$\text{Time}_{\text{Mem\_New}} = \frac{\text{Bandwidth}_{\text{Mem\_New}}}{\text{Bandwidth}_{\text{Mem\_Known}}} \times \text{Time}_{\text{Mem\_Known}}$$

Design--Communication Simulation

- NCCL (NVIDIA Collective Communications Library) 会动态选择最优的算法将**集合通信操作**（AllReduce，AllGather）**转换为点对点操作**send/receive。
- **问题：**从头复现NCCL算法选择和转换过程十分困难。
- **SimCCL**
  - NCCL的修改版，以拦截NCCL关键操作。
  - 单台主机上运行，使用“劫持”技术来捕获NCCL生成的P2P列表。
  - 通过动态链接库注入的方式来拦截和修改原始NCCL库的行为。

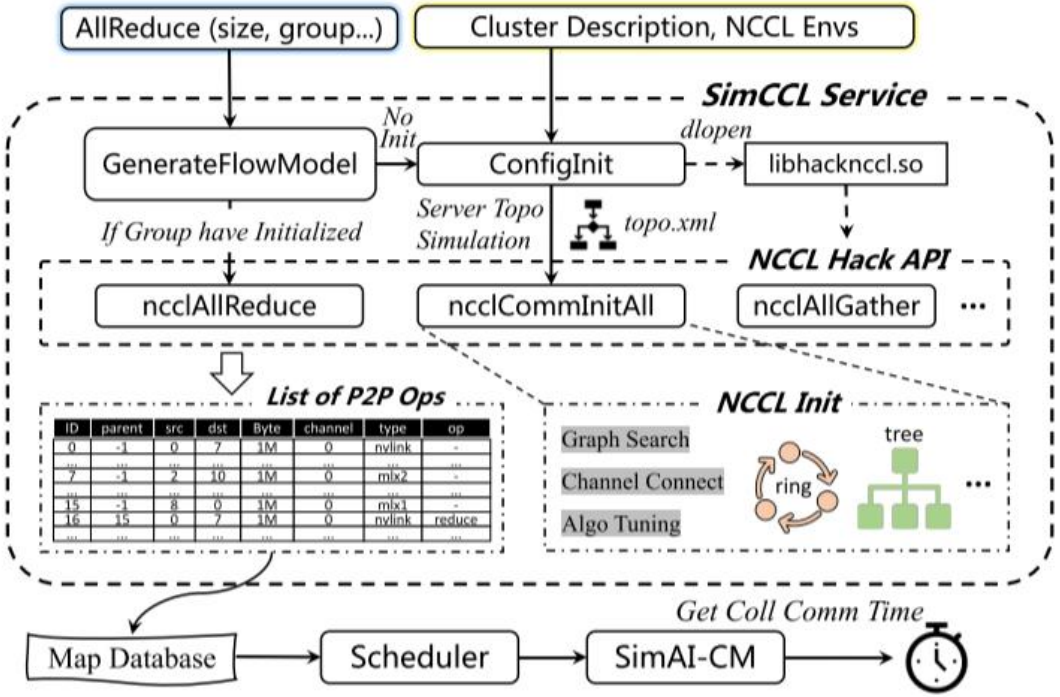


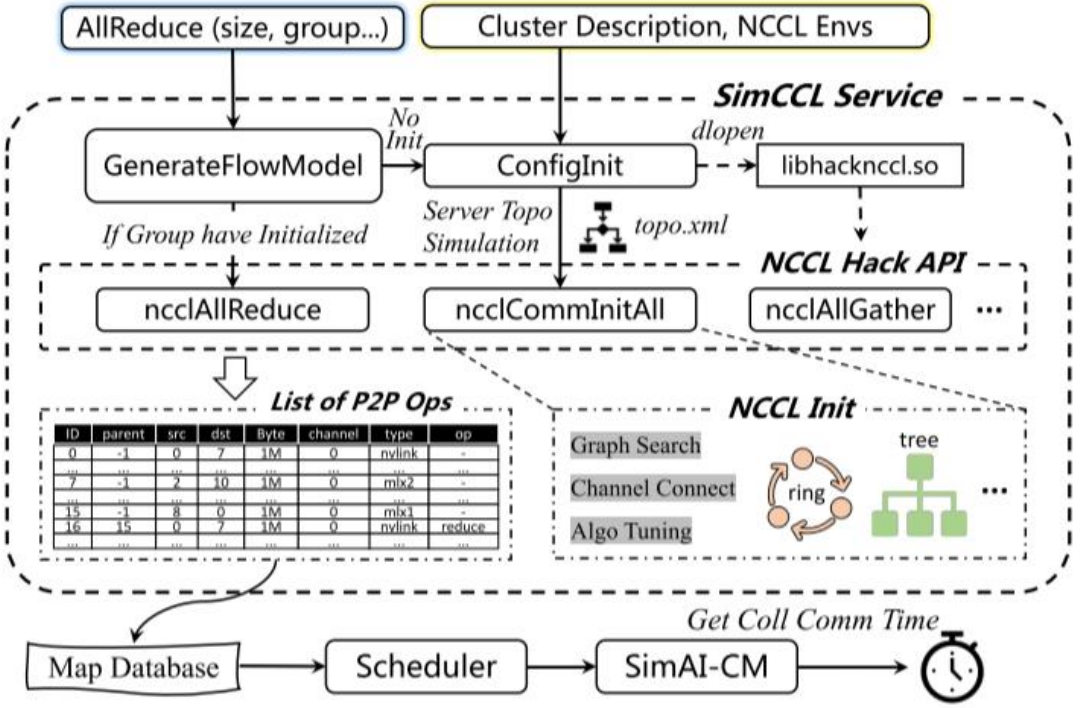
Figure 3: Illustration of how SimAI transforms a collective communication operation to a list of peer-to-peer communication operations.

Design--Communication Simulation

□对NCCL行为的关键修改：

- NCCL初始化：拦截ncclCommInitAll以创建虚拟通信器，使NCCL“认为”自己运行在一个真实集群环境中。
- 读取配置：读取用户指定的拓扑文件，而非实际设备搜索。
- 集体通信转换：拦截集体通信调用，捕获这些描述底层通信事件的信息，而不进行实际数据传输。

□NCCL参数支持：支持绝大多数NCCL参数，包括PCI✖NVLink (PXN) 等用于优化轨道拓扑的特性。



ID	parent	src	dst	Byte	channel	type	op
0	-1	0	7	1M	0	nvlink	-
...	...	...	...	...	...	...	...
7	-1	2	10	1M	0	mlx2	-
...	...	...	...	...	...	...	...
15	-1	8	0	1M	0	mlx1	-
16	15	0	7	1M	0	nvlink	reduce
...	...	...	...	...	...	...	...

Design--Simulation Speed up

口多线程加速

- 为SimAI-CM实现，以应对超过千卡的模拟。
- 选择UNISON（Eurosys, 2024），因其开源、自动拓扑分区和可扩展性。

Unison: A Parallel-Efficient and User-Transparent Network Simulation Kernel

EuroSys'24

Songyuan Bai, Hao Zheng, Chen Tian, Xiaoliang Wang, Chang Liu, Nanjing University;

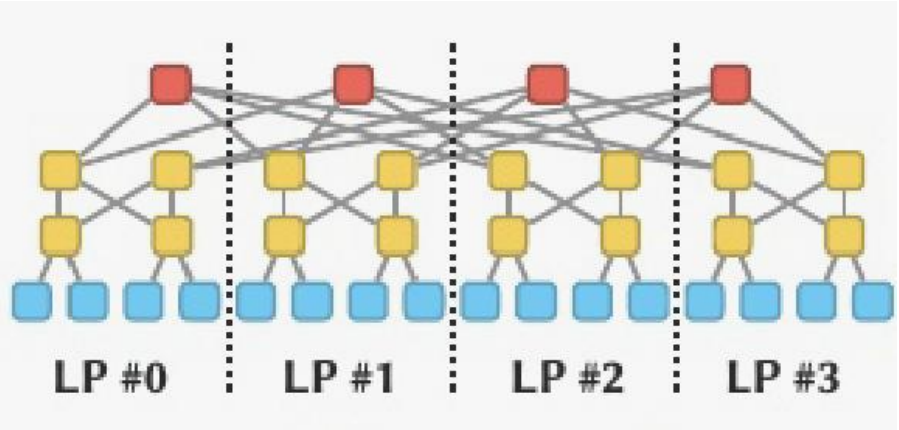
Xin Jin, Peking University; Fu Xiao, Nanjing University of Posts and Telecommunications;
Qiao Xiang, Xiamen University

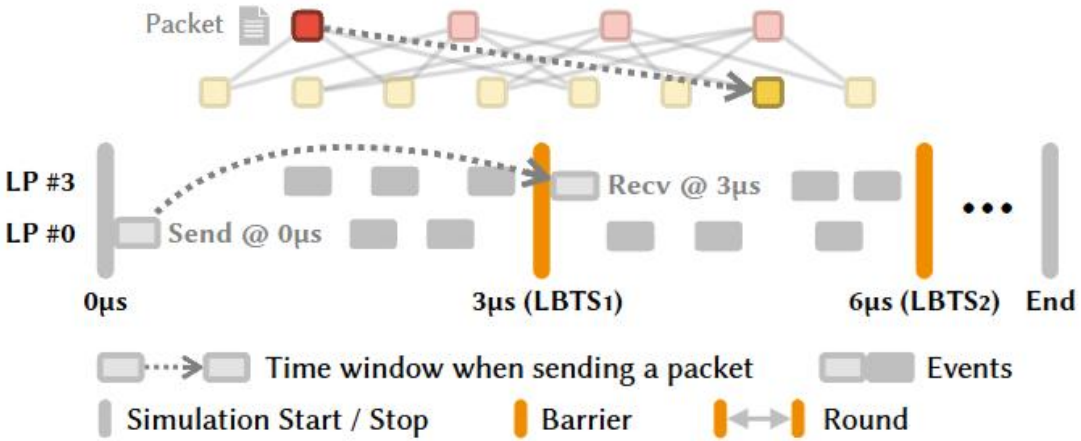
Design--Time Window

□原先的单线程离散事件仿真器

```
initialize -- 包括构建模型，添加初始化事件到事件堆FES中
while (FES not empty && simulation not yet complete)
{
    retrieve first event from FES
    t := timestamp of this event
    process event
    (processing may insert new events in FES or delete existing ones)
}
finish simulation (write statistical results, etc.)
```

□基于屏障同步算法的多线程仿真器

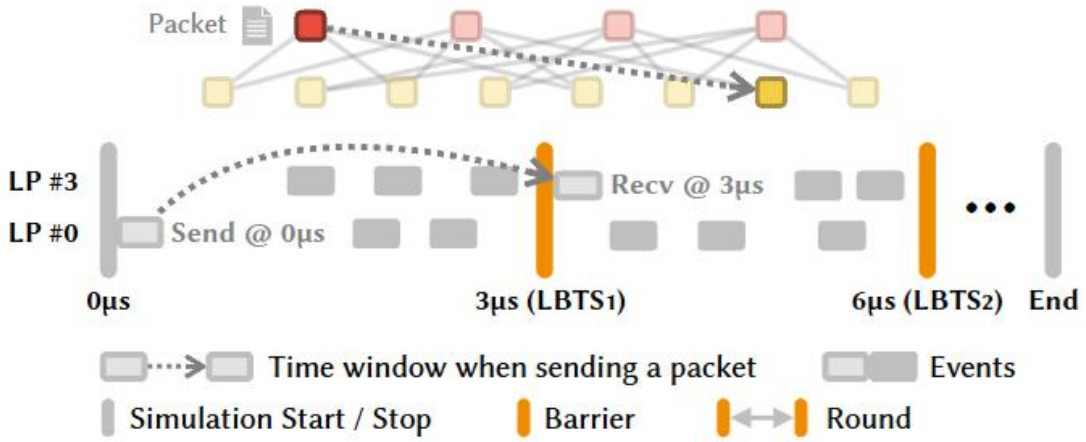




Design--Time Window

口实现改进的屏障同步算法

- 时间窗口设置允许LP在一定的时间范围内自由处理事件，而不会影响其他LP的处理。
- Lookahead value是设置时间窗口的关键参数。它定义了LP在处理事件时可以“向前看”的最大时间范围。它基于LP之间连接的最小链路延迟。
- 避免因果倒置的关键是：lookahead机制确保了LP不会收到来自其他LP的“未来事件”！



每个LP计算其下一个事件的时间戳，并根据lookahead value确定其当前时间窗口的结束时间（Lower Bound on Time Stamp, LBTS）。LBTS的计算公式为：

$$\mathrm{LBTS} = \min\left(\mathrm{LBTS}_{\mathrm{pub}}, \min\{\mathrm{next\_event\_time}_i\} + \mathrm{lookahead}\right)$$

其中：

- $\mathrm{LBTS}_{\mathrm{pub}}$是公共LP的下一个事件的时间戳。
- $\mathrm{next_event_time}_i$是第$i$个LP的下一个事件的时间戳。
- lookahead是前瞻值。

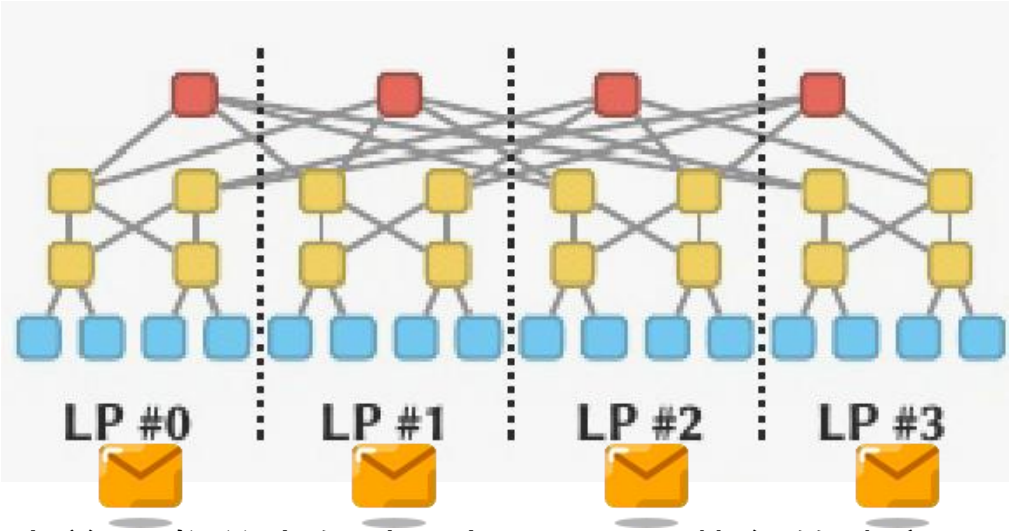
Design--Cross LP Event

□事件传递

- 使用邮箱（mailbox）机制来缓存这些跨LP事件。发送LP将事件放入接收LP的邮箱中，而不是直接插入接收LP的FEL。

□事件接收

- 事件接收：在接收事件阶段，接收LP会从邮箱中取出所有跨LP事件，并将它们插入到自己的FEL中。这些事件会根据时间戳排序，确保因果一致性。

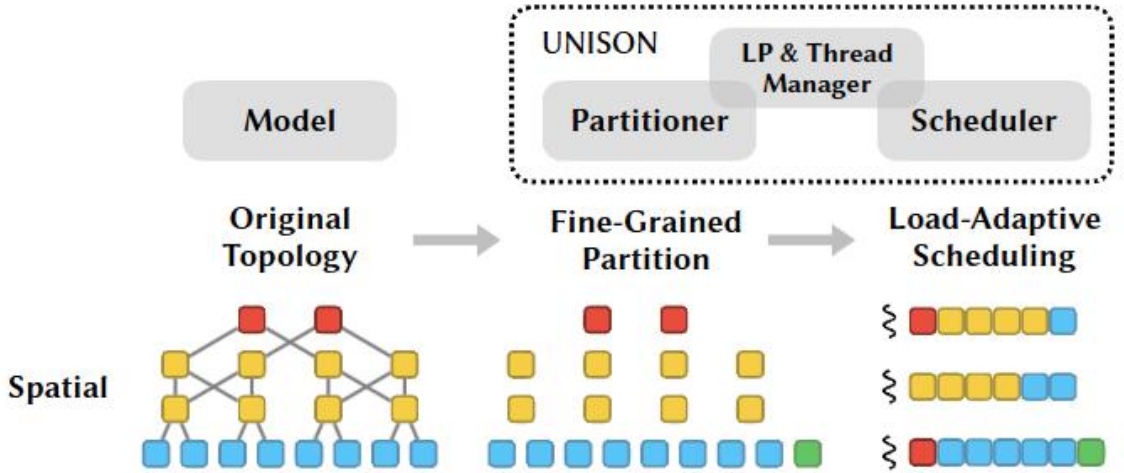


避免当其余多个LP向当前LP发送消息时，由于不同LP执行的速度不同导致时间戳靠后的消息被先传给当前LP直接插入消息队列中，从而产生因果倒置。

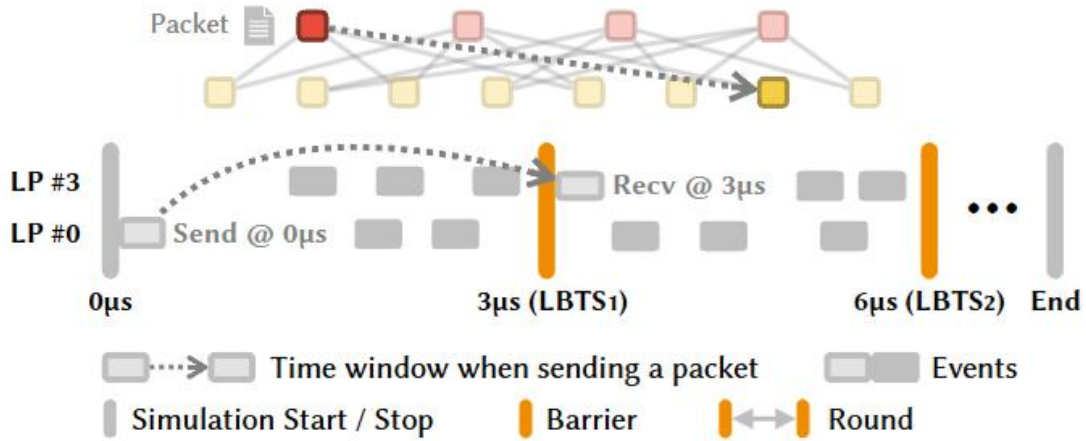
Design

并行仿真执行流程

- **初始化阶段**
  - 1、网络分区
  - 2、线程启动与资源分配
    - 每个LP绑定到一个独立的线程
    - 初始化本地事件队列、邮箱
- **周期性时间窗口处理（四个阶段）**
  - Phase 1: 处理本地事件（并行执行）
    - 判断：本地事件or跨区事件
  - Phase 2: 处理全局事件
  - Phase 3: 合并跨区消息（并行执行）
  - Phase 4: 同步计算新的时间窗口



屏障同步机制



Design--Simulation Speed up

□全局变量的无锁共享

- 问题：共享数据结构（主机间通信元数据、交换机队列长度）上的全局锁导致性能瓶颈。
- 解决：通过以线程无关的方式（节点ID索引表）管理这些变量，重构SimAI以在无全局锁的情况下运行。

Evaluation--Testbeds and Benchmarks

测试平台

- 两个集群（A100 & H100），多轨道、胖树、主机间RoCEv2（alibabaHPN，SIGCOMM24）。
- **Cluster A100**: 128台主机，每台配备8个A100 GPU，4个Mellanox ConnectX-6网卡（2x100Gbps）。NVLink 600GBps。
- **Cluster H100**: 128台服务器，每台配备8个H100GPU，8个Mellanox ConnectX-7网卡（2x200Gbps）。NVLink 900GBps。

基准测试

- 模型及参数源自SimAI基准测试套件。
- 对比平台：ASTRA-sim（通过工作负载适配器增强，以解析SimAI-WG生成的工作负载文件）
- 工作负载文件的模型、框架和NCCL参数与真实物理集群任务相一致。
- 评估的集群规模：128、512、1024个GPU。
- 展示的模型：GPT-3 13B、LLaMA 65B、GPT-3 175B。

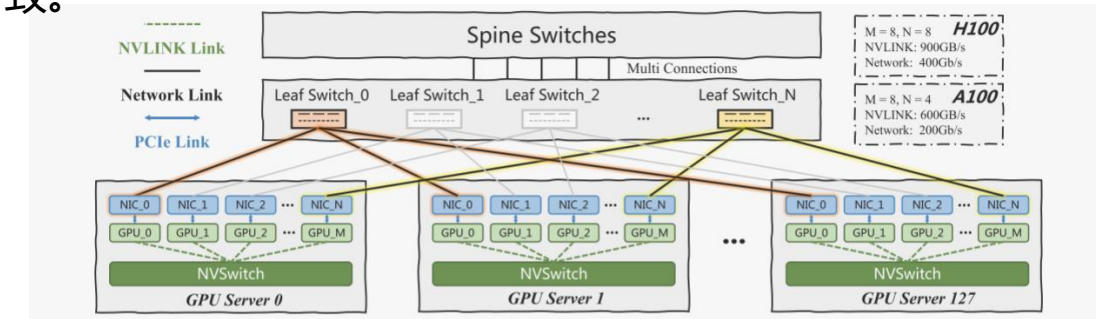


Figure 5: Multi-rail network topology of cluster H100. For cluster A100, two GPUs share a single NIC

Evaluation--Benchmark suite

□真实工作负载代表

- 分析过去六个月的模型参数规模与任务规模
- 超过94%的LLM是GPT-3或LLAMA的变体

□实用性

- 选择一个最小化的基准集合，覆盖典型配置
- 包含一系列不同大小的开源LLM模型

□影响工作负载模式的关键参数

- 模型参数：hidden_size, num_layers, seq_len...
- 框架参数： world_size, zero_level, reduce_bucket_size, 并行策略...

Model		Model_hyperparameter				Framework	Parallel Parameter		Ds_config					
Name	Parameter size	Hidden size	Layers	Sequence length	FFN size	Name	TP	PP	Zero level	Reduce bucket size	Allgather bucket size	Prefetch bucket size	Max live parame-ters	Param persistence threshold
GPT-3 6.7B	6.7B	4096	32	2048	16384	Megatron	1	1	-	-	-	-	-	-
GPT-3 13B	13B	5120	40	2048	20480	Megatron	2	1	-	-	-	-	-	-
GPT-3 175B	175B	12288	96	2048	49152	Megatron	8	8	-	-	-	-	-	-
LLaMA 65B	65.2B	8192	80	4096	28672	Megatron	8	2	-	-	-	-	-	-
Llama3 405B	405B	16384	126	8192	53248	Megatron	8	16	-	-	-	-	-	-
LLaMA 7B	6.7B	4096	32	4096	11008	Deepspeed	1	-	2	1.00E+09	1.00E+09	-	-	-
LLaMA 65B	65.2B	8192	80	4096	28672	Deepspeed	1	-	3	1.00E+09	-	1.00E+09	6.00E+08	1.00E+06

Evaluation--Precision of Communication Simulation

- **主机内通信**
  - SimAI比ASTRA-sim精确得多，与真实值高度吻合。
  - 平均偏差 (SimAI vs ASTRA-sim): A100 (3.9% vs 74.8%), H100 (2.3% vs 51.7%)。
  - ASTRA-sim表现不佳，尤其是在AllGather/ReduceScatter操作上 (例如，A100 8GB 消息误差 >21.8%，在不利条件下误差率高达 210.5%)。

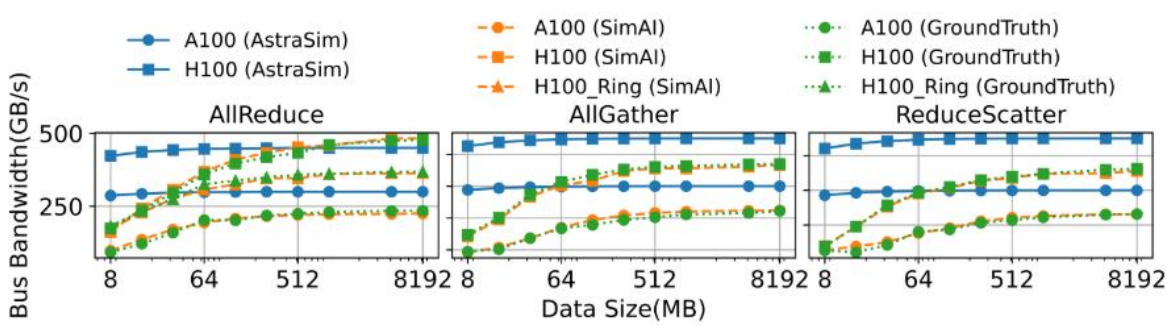


Figure 6: Bus bandwidths of intra-host collective communication operations vary with different message sizes and GPU types. Note that the performance is not related to the cluster scales.

Evaluation--Precision of Communication Simulation

- 主机间通信
  - ASTRA-sim难以**准确模拟小消息通信**，而SimAI始终接近真实值。
  - 随着通信规模的扩大，ASTRA-sim的模拟与真实值之间的差异也随之增大 (例如，AllGather 8MB，128 个 A100 GPU：仿真时间误差 45.9%；规模扩大到 512个GPU 时，误差增至 530.2%)。

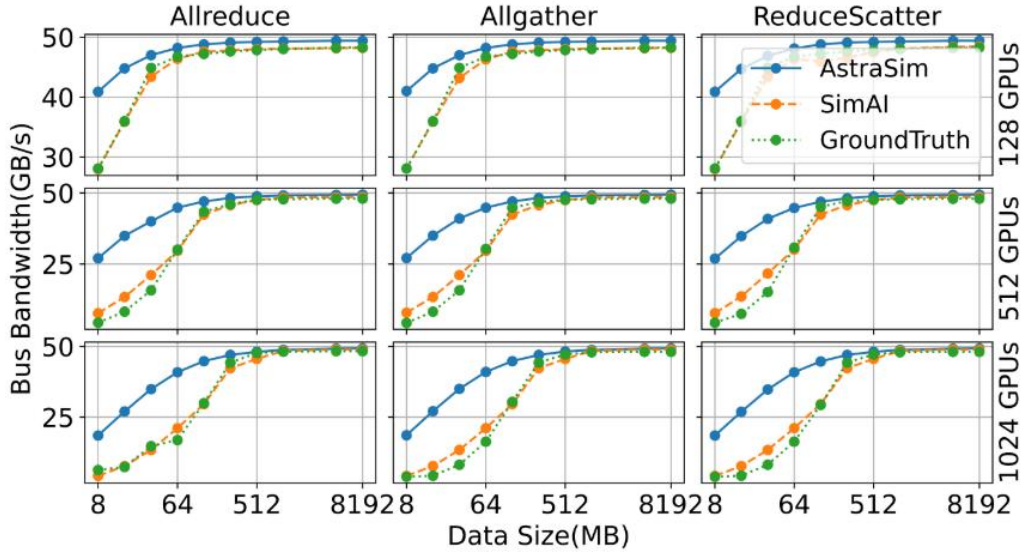


Figure 7: Bus bandwidths of inter-host collective communication operations with different message sizes and cluster scales.

Evaluation--Precision of Computation Simulation

- 测试了SimAI-CP和SimAI-CP-Model相对于真实值 (Nvidia A100, H100, H20) 和ASTRA-sim (仅使用 SCALE-sim 进行矩阵乘法) 的精度。
- SimAI-CP: 高度准确，总执行时间差距极小 (0.5%-3.1%)。
- SimAI-CP-Model: **偏差较大 (13%-15%)，建议在GPU不可用时使用**。
- ASTRA-sim (SCALE-sim): 存在显著误差 (H100: 49.8%, A100: 117.9%, H20: 224%)。
- 也尝试了 Accel-Sim，但由于PyTorch版本过旧而失败。

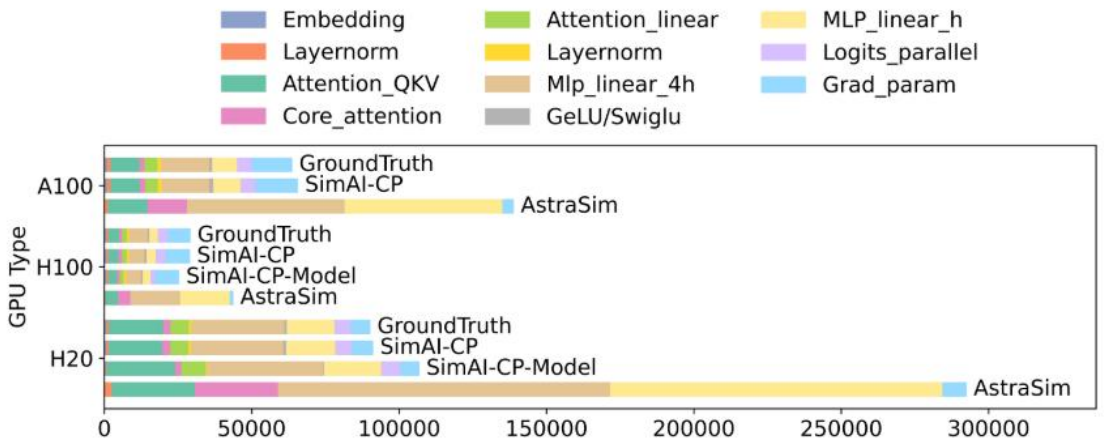


Figure 8: The execution time of the computation kernels of GPT-3 175B on different GPUs.

Evaluation--Precision of End-to-end Simulation

- SimAI: 所有工作负载的迭代时间与真实集群中实际LLM任务的迭代时间非常接近。
  - 即使在1024个GPU的规模下，差距也小于4%。
  - 与真实值的偏差小于3.9%，比ASTRA-sim精确36.1倍。
  - 随着模型规模的增加 (例如，从GPT-3 13B到GPT-3 175B)，SimAI的精度相应提高 (因为GPT-3 175B的**集体通信操作的消息规模更大，其模拟更为精确**)。
- ASTRA-sim: 对于A100和H100 GPU的迭代时间均不精确。

In Production Benefits--Guiding Host Design for New GPUs

- **H100 GPU 场景:**
  - 问题：新的H100 GPU需要多少**网络带宽**才能**最大限度地发挥性能并最大限度地降低成本**?
  - SimAI 模拟了具有100Gbps、200Gbps和400Gbps网络配置的1024个H100 GPU。
  - 结果：从 200Gbps 升级到 400Gbps 可带来19%的性能提升，证明了成本的合理性。这为最终的主机设计 (8个H100, 8个2x200Gbps 网卡) 提供了依据。
- **H20 GPU 场景:**
  - H20的计算性能低于H100。
  - SimAI结果：网络带宽从100Gbps增加到 200Gbps，端到端训练性能提高11%；从200Gbps增加到400Gbps，仅提高 6%。
  - 决策：为每个H20配备200Gbps的网络带宽。
  - 即使对于未发布/缺货的 GPU 设备，SimAI-CP-Model 也能提供有意义的结果。

In Production Benefits--Quantifying Scaling-up Benefits

- 挑战: 在实际训练集群中探索所有**参数组合 (尤其是并行化策略)** 会产生巨大的开销。
- SimAI解决方案：自动识别最佳参数设置。
- **张量并行（TP）配置研究**:
  - 传统观点：TP组大小等于通过NVSwitch连接的主机内的GPU数量。
  - SimAI实验：GPU数量为8、16、32 和 64，均通过NVSwitch互连。
  - 主要发现:
    - 在8个GPU主机中，GPT-3 13B、LLaMA 65B和GPT-3 175B的最佳TP大小分别为 4、8 和8。
    - 对于任何特定的TP大小，主机中GPU越多，性能越好。
    - 即使GPU数量更多，如果TP大小不合适，性能也可能下降。
  - 实践指南:
    - TP参数应适应整个模型层。然而，增加TP大小会降低端到端吞吐量；更好的策略是设置更多的数据并行 (DP) 组进行并行训练。
    - GPU主机设计必须考虑LLM的发展特性。如果已知层的最大大小，则应相应地确定主机内的GPU数量，优先考虑增强横向扩展性能。

Conclusion

□统一性 (Unification)

- 整合了训练框架、核心计算和集合通信库的关键流程， 实现在单一框架内进行架构设计和性能调优 。

□高精度 (High Precision)

- 通过“劫持”训练框架与NCCL，生成精确的工作负载与点对点通信流量；通过在目标GPU实测计算操作维护一个kernel级计算任务的历史数据库。

□可扩展 (Scalability)

- 一个完整的工具集。可以分析集合通信的RDMA流量特征、在非GPU的集群进行网络测试分析、进行workload的分析演算、进行workload的全流程仿真。

Thinking

□能否提高

- overlap：计算/通信、通信/通信
- 显存分析：确保正确参数输入
- 多租户集群：合理分配资源；故障模拟/恢复；负载均衡与流量冲突

□与此类似

- 机器学习模型替代：某些计算密集型或重复性高的模拟组件or有规律的集合通信操作
- 借鉴编译器优化技术，对工作负载进行探索，评估其对资源利用率
  - 模拟硬件资源（如SM、通信链路），对DAG进行不同的调度策略模拟
  - 模拟将某些相邻算子融合或将大算子分解对计算时间和内存访问模式的影响

□问题泛化

- 支持推理框架

Q & A

Presenter: Botai Sun