arXiv:2603.26823v1 [cs.LG] 27 Mar 2026

大规模人工智能系统中的吞吐量优化：来自数据加载器与内存分析创新的证据

Mayank Jha Amazon
Robotics
mayankjh@amazon.com

摘要—**大规模基础模型，尤其是大语言模型（LLM）的发展，受到显著的计算和内存瓶颈制约。这些挑战使吞吐量优化不再只是单纯的工程任务，而成为关键的战略杠杆，直接影响训练时间、运营成本以及下一代模型可实现的规模。本文综合近期学术界和工业界创新成果的证据，分析训练效率方面的关键进展。我们考察了针对数据加载器瓶颈的架构解决方案，例如 OVERLORD 框架，该框架已实现端到端训练吞吐量提升 4.5×。我们研究了旨在突破 GPU 内存墙的内存优化技术，包括 DeepSpeed 的 ZeRO-Offload 等 CPU 卸载策略，这些策略使训练远超单加速器容量的模型成为可能。此外，我们还探讨了以编译器为中心的优化日益增长的重要性，Triton-distributed 便是其中的代表，它能够联合优化计算、内存和通信，从而显著提升性能。本文还结合高级性能分析工具和硬件特性研究进行分析，这些工具和研究能够识别并缓解此前被忽视的开销，例如动态电压与频率调节（DVFS）带来的开销。研究结果表明，必须采取全面的系统级方法，整合数据流水线、内存管理、网络结构和编译器技术等领域的创新，才能加速人工智能发展、控制成本，并拓展模型规模的边界。**

索引词—**大规模人工智能系统、基础模型、吞吐量优化、图形处理器内存、数据加载器瓶颈、分布式训练、编译器优化、Triton、内存分析**

I. 引言

人工智能领域日益由大规模基础模型主导，例如 Google 的 Gemini 和 Meta 的 LLaMA 系列。这些模型在海量数据集上进行训练，并可适配多种下游任务 [1]。训练这些模型是一项艰巨的工程挑战，需要配备专用超级计算机集群，其中包含数万乃至数十万台加速器。例如，OpenAI 训练 GPT-4 时使用了 Microsoft Azure 上的数万台 GPU，而据报道，Meta 正在一个拥有超过 100,000 个 NVIDIA H100 GPU 的集群上训练 LLaMA 4 模型 [1]。如此庞大的规模会引发数据输入、加速器内存容量和节点间通信方面的严重瓶颈，这些问题共同阻碍性能并推高运营成本。下一代系统（如 NVIDIA GB200 NVL72）的总拥有成本约为 H100 系统的 1.6 倍 [2]。

因此，吞吐量优化已成为开发此类系统的组织所采用的首要战略杠杆。提高训练效率可直接转化为三项关键的商业与研究优势：（1）缩短训练耗时，从而加快研究与开发周期；（2）降低运营成本，因为单次训练所需的计算支出可能高达数百万美元；（3）突破根本性的硬件内存限制，进而训练规模更大、能力更强的模型。本文旨在综合分析人工智能训练技术栈近期涌现的高影响力创新——涵盖从数据加载和内存管理到编译器技术与网络优化的各个方面——以解决这些关键瓶颈。通过考察具体工具、架构模式和研究成果，本文阐明了系统性的吞吐量优化如何推动大规模人工智能持续扩展与进步。

II. 文献综述

人工智能训练扩展的主要障碍是众所周知的硬件限制，通常称为“墙”。最直接的是*GPU 内存墙*：单个加速器的显存容量不足以容纳训练所需的模型参数、梯度和优化器状态，这是一个显著的瓶颈 [3]。对于使用 Adam 优化器的基于 Transformer 的模型，一种常见的估算方法是，所需显存约为模型参数量（以十亿为单位）的 40 倍 [4]。这意味着，一个相对小型的 70 亿参数模型至少需要 280GB 显存，远超单个最先进加速器的容量，因此必须采用分布式训练范式 [4]。

表 I 总结了具有代表性的模型规模所需显存的估算值，展现了 GPU 内存墙的严峻程度。

除了本地内存，数据移动本身也被认为将成为瓶颈。随着计算能力（以浮点运算次数计量）的增长，网络和内存子系统为处理单元供给数据的能力将成为限制因素。研究项目预测，由于数据移动方面的限制，对于为期三个月的训练任务，当计算量超过 2×10^{28} FLOPs 时，高效扩展可能变得不可行 [5]。更根本的是，

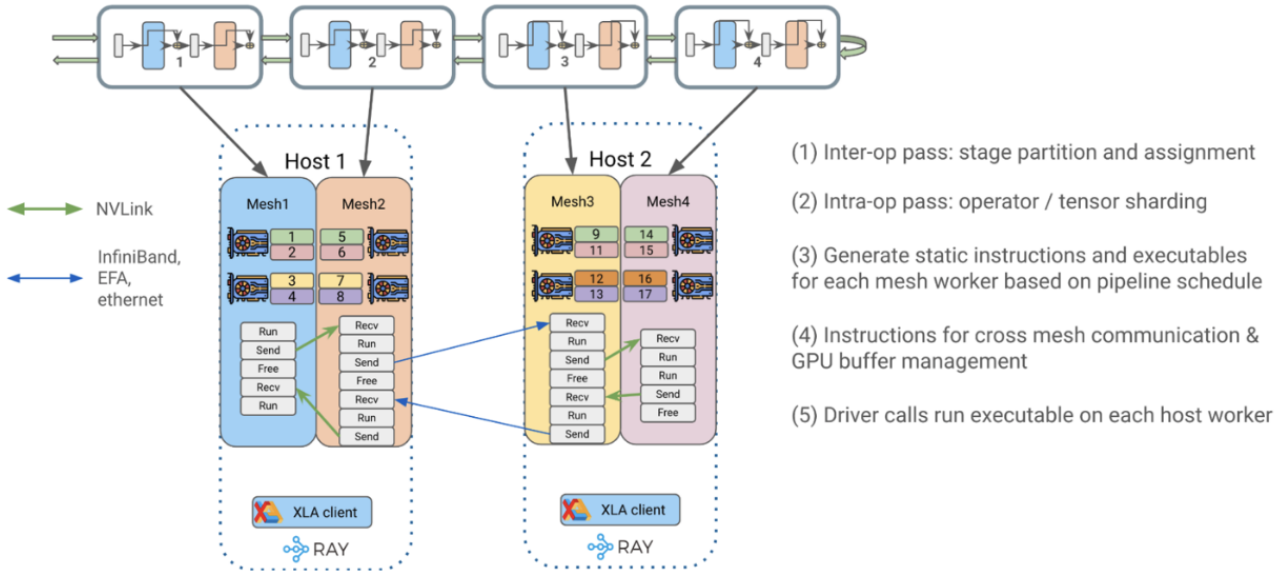

图 1. 系统级执行模型，展示跨主机的流水线阶段划分、每台主机内部的算子内张量分片，以及通过高速互连进行的跨主机通信。静态执行调度会生成每工作器指令，用于协调计算、通信和 GPU 缓冲区管理，从而揭示大规模训练中吞吐量和成本瓶颈的主要来源。

表 I 使用 Adam 优化器训练 Transformer 模型的预计显存需求（≈40×参数量）

模型规模	预计显存	GPU（80 GB）
1B	40 GB	1
7B	280 GB	4
13B	520 GB	7
70B	2,800 GB	35
175B	7,000 GB	88

在约 2×10^{31} FLOPs 时存在一道“延迟墙”：此时，完成单个梯度步所需的时间会短于节点间通信的网络延迟。由于延迟的改善速度慢于其他硬件指标，这一挑战尤其难以克服 [5]。这些理论和实际限制推动了多样化的软件、硬件和算法解决方案生态的发展，旨在管理内存层次结构、优化数据流水线，并在大型分布式系统中高效协调通信。

III. 方法论

本文对聚焦于大规模人工智能训练系统吞吐量优化的当代研究与技术文献进行了定性综合。方法论包括系统性审阅同行评审文章、人工智能实验室发布的技术报告，以及2024年至2025年间发表的行业分析。筛选标准优先考虑那些在现实环境或大规模仿真环境中已证明能够带来可量化性能提升的创新。分析围绕训练流水线中的关键瓶颈展开：数据加载、图形处理器内存管理、分布式通信和硬件利用率。通过整合学术研究（例如编译器设计）、开源工具（例如DeepSpeed）以及大型科技公司（例如Meta、英伟达）的架构最佳实践，本研究全面呈现了当前用于提升训练效率的战略杠杆。

IV. 研究发现与分析

本节综合近期研究成果与系统级证据，分析大规模人工智能训练系统中数据流水线、内存管理、分布式通信以及面向硬件的性能调优等方面的关键吞吐量优化策略。

A. 数据流水线管理创新

数据加载，即将训练样本从存储提供给加速器的过程，已成为一个重要瓶颈，尤其是在使用多源数据集和复杂并行策略时。OVERLORD 架构通过将数据预处理流水线拆分为专用执行单元来解决这一问题：由“源加载器”负责样本级转换，由“数据构造器”负责批次级操作 [6]。这一设计结合集中式数据平面，消除了冗余数据访问并降低了内存开销。在数千个图形处理器组成的集群上进行生产部署时，OVERLORD 使端到端训练吞吐量提升了 4.5× ，并将 CPU 内存使用量降低了 13.5× ，这表明优化数据流水线是整体系统性能的关键组成部分 [6]。

B. 克服 *GPU* 内存墙的策略

为突破单个图形处理器的物理内存限制，人们开发了各种卸载技术。

表 II 图形处理器内存优化策略比较

策略	机制	优势	权衡
ZeRO 阶段 1	卸载优化器状态到中央处理器	中等显存节省	开销低
ZeRO 阶段 2	卸载梯度 + 优化器	High VRAM 节省	中等开销
ZeRO 第 3 阶段	将所有参数卸载到中央处理器	训练 10B+ on 1 图形处理器	更高通信成本
激活值卸载	将激活值移至 CPU内存	降低 peak VRAM	Sync 开销
统一内存-ory	一致性 CPU–图形处理器地址空间	透明迁-移	硬件相关

DeepSpeed 的 ZeRO-Offload 是一个典型示例，它将内存和计算任务交由主机中央处理器和随机存取存储器处理，使得在单个图形处理器上训练参数量超过 100 亿的模型成为可能 [7]。它的不同阶段允许逐步将优化器状态、梯度乃至模型参数卸载到系统内存中，从而释放宝贵的显存 [7]。一种类似的技术是将激活卸载到中央处理器，即把中间激活张量暂时移至 CPU内存 [8]。虽然这种方法效果显著，但会带来同步开销，进而影响图形处理器利用率。在英伟达 Grace Hopper 系统等专用硬件上，统一内存可在中央处理器与图形处理器之间提供一致的内存空间，不过其效率取决于工作负载：某些微调任务的开销极小，而其他任务则会发生大量数据迁移 [8]。这些策略不仅是优化手段，更是赋能技术，使得训练原本会因内存限制而无法训练的模型成为可能。

表 II 比较了主要的内存优化策略及其特征。

C. 面向分布式工作负载的以编译器为中心的优化

近期进展已将优化工作的重点转向编译器层面，从而能够更精细地控制硬件资源。Triton-distributed 是 Triton 编译器的扩展，也是首个支持分布式人工智能工作负载原生重叠优化的工具 [9]。它将符合 OpenSHMEM 标准的通信原语集成到高级 Python 编程模型中，从而能够联合优化计算、内存访问和通信 [9]。与采用 NCCL/RCCL 的标准 PyTorch 基线相比，这种方法实现了 1.09× 至 44.97× 的加速比 [9]。该框架的编程模型遵循多程序多数据（MPMD）范式，运用“对称内存”和“异步任务”等概念来协调并行操作 [9]。

D. 网络结构与通信优化

在大规模场景下，网络性能至关重要。Meta 发现，为 RoCE 调优标准 DCQCN 拥塞控制

表 III 互连对 12B 参数模型微调的影响

互连	带宽	单步时间	加速比
10 吉比特以太网	∼1.25 GB/s	39.8 秒	1×
IB Quantum-2（400G）	∼400 GB/s	4.4 秒	9×

（融合以太网上的远程直接内存访问）对于训练工作负载而言颇具挑战，因此他们采用了优先级流量控制（PFC）以及协同设计的接收方驱动流量准入控制机制 [10]。为改善负载均衡，他们还通过对 RoCE 队列对（QP）字段进行哈希，实现了增强型等价多路径（E-ECMP），使全归约性能最高提升了 40% [10]。互连技术的选择也至关重要；基准测试表明，在微调一个 12B 参数模型时，采用英伟达 Quantum-2 InfiniBand（∼400GB/s）相比标准 10Gbit 以太网可使训练加速比达到 9×，并将每步耗时从 39.8 秒缩短至 4.4 秒 [11]。

表 III 总结了互连技术对分布式训练性能的影响。

E. 高级性能分析与硬件特定调优

要找出性能差距的根本原因，需要使用先进的性能分析工具。使用“Chopper”性能分析工具对超威半导体 InstinctTM MI300X 节点进行分析后发现，在 Llama 3 8 B 训练期间，动态电压频率调整（DVFS）带来的频率开销，是理论性能与实际观测性能之间差距的最大单一因素 [12]。同一项研究还发现，FSDPv2（一种分布式训练策略）的内存分配更具确定性，使 GPU 能够维持比 FSDPv1 高约 20%、且更加稳定的时钟频率，其收益超过了额外数据复制操作带来的开销 [12]。这凸显了深入开展硬件感知性能分析的重要性，它有助于发现那些不易察觉的优化机会。

F. 异构硬件抽象

随着 AI 基础设施日益多元化，不再局限于单一厂商，硬件抽象对于可移植性和效率至关重要。UniOrch 框架通过采用基于 Triton 和 MLIR 的统一编译系统来应对这一挑战 [13]。它会捕获 PyTorch 算子调用，并将其转换为独立于硬件的中间表示（Triton-IR）。随后，该中间表示会经过优化并降级为特定硬件的后端，例如英伟达的并行线程执行（PTX）或华为的 CANNIR，从而让同一代码库能够在不同的 GPU、神经网络处理器和深度计算单元上高效运行 [13]。这种方法能够减少开发开销，并使组织能够充分利用多厂商加速器生态系统。

表 IV 汇总了本文分析的主要吞吐量优化创新。

V. 讨论

这些发现共同展现了大规模人工智能训练中的明显趋势：优化正从孤立的

表 IV 大规模人工智能训练中吞吐量优化创新成果汇总

优化领域	创新方法／工具	关键机制	报告的改进效果
数据流水线	OVERLORD [6]	解耦执行器，集中式数据平面	4.5× 吞吐量，13.5× 内存缩减
内存管理	ZeRO-Offload [7]	渐进式 CPU/RAM 卸载	在 1 个图形处理器上训练 10B+ 模型
编译器	Triton-distributed [9]	计算/通信重叠的多程序多数据	1.09×–44.97× 通过 PyTorch+NCCL
网络结构	E-ECMP（Meta） [10]	基于队列对的哈希实现负载均衡	AllReduce 性能最高提升 40%
互连	InfiniBand 400G [11]	高带宽 RDMA互连	9× 相较于 10G 以太网的训练加速比
性能分析	Chopper [12]	多级图形处理器特性分析	发现动态电压频率调整是首要性能瓶颈
硬件抽象	UniOrch [13]	Triton/MLIR 统一编译	适用于 GPU/NPU/深度计算单元的单一代码库

组件级改进转向整体的系统级协同设计方法。吞吐量的战略重要性显而易见，因为它直接助力实现缩短训练时间、降低运营成本以及支持更大模型规模等主要目标。OVERLORD 等创新表明，数据流水线曾经只是次要考量，如今却可能成为主要瓶颈，因此也是极具价值的优化领域 [6]。同样，ZeRO-Offload 等内存卸载技术不再只是为了节省成本；它们是能够从根本上改变现有硬件上模型规模限制的赋能技术 [7]。

以 Triton-distributed 为代表的编译器中心化解决方案的兴起，标志着该领域迈入了一个重要的成熟阶段 [9]。这些工具为开发者提供高级抽象，以控制低层级的硬件特定操作（例如让通信与计算重叠），从而普及了性能工程，并释放出仅靠基于库的方法难以实现的性能提升。Meta 的网络工程实践也为此提供了进一步佐证：要提升性能，必须协同设计网络架构配置（E-ECMP）与集合通信库 [10]。其实际意义重大：要实现最先进的性能，必须在硬件、网络、编译器和应用层之间进行深度集成。

当前领域的一个主要局限是专业化程度过高。许多先进技术都针对特定的硬件互连进行了定制，例如，Triton-distributed 针对英伟达的 NVSwitch 与超威半导体的全网状拓扑采用了各不相同的“交错排列”优化 [9]。这种定制化虽然能够充分发挥性能，但也可能增加软件复杂性并阻碍可移植性。UniOrch 等框架旨在通过硬件抽象层缓解这一问题，但性能与可移植性之间往往仍需权衡 [13]。此外，尽管本文聚焦于吞吐量，但模型准确率、收敛速度和训练能效等因素同样是至关重要的维度，需要进行更全面的分析。性能分析工具的发现指出，动态电压频率调整（DVFS）是一个主要的性能抑制因素；这表明，通过缩小理论硬件能力与实际应用性能之间的差距，仍可实现显著且不易察觉的性能提升 [12]。

六、结论

本文探讨了在大规模 AI 时代，吞吐量优化作为一项关键战略杠杆的重要作用。证据表明，解决整个训练技术栈中的瓶颈至关重要：从利用 OVER-LORD 等系统提升数据摄取效率，到运用 ZeRO-Offload 等技术应对内存限制，再到借助 Triton-distributed 等编译器级框架提高通信效率。这些创新并非渐进式改进，而是推动发展的基础性支撑，能够缩短上市时间、控制不断攀升的计算成本，并支持探索规模更大、能力更强的基础模型。

分析强调，采用整体性、系统感知的方法至关重要，需对硬件、软件和网络进行协同设计，以实现最高效率。未来研究应聚焦于进一步整合这些彼此独立的优化，或可借助更先进的编译器生态系统，自动执行硬件特定调优，并抽象掉异构系统的复杂性。此外，开发与编译器集成的可移植性能分析工具，使其能够自动识别并缓解动态电压频率调整（DVFS）等细微的性能抑制因素，对于确保下一代硬件的理论潜力在实践中得到充分发挥至关重要。

参考文献

[1]P. Tiwari，《为大语言模型（LLMs）设计并适配基础设施规模——第 2 部分：案例研究》，*Medium*，2025年5月。 [在线]。可访问：https://medium.com/@deep.bbd/designing-and-right-sizing-infrastructure-for-large-language-models-llms-part-2-case-s

[2] D. Patel 和 D. Nishball，《H100 与 GB200 NVL72 训练基准：功耗、总拥有成本、可靠性分析及软件随时间的改进》，SemiAnalysis，2025年8月。 [在线]。链接：https://newsletter.semianalysis.com/p/h100-vs-gb200-nvl72-training-benchmarks [3] S. Rajbhandari、J. Rasley、O. Ruwase 和 Y. He，《ZeRO：面向万亿参数模型训练的内存优化》，载于国际高性能计算、网络、存储与分析会议（SC20）论文集，第 1–16 页，2020年。[4]俄亥俄超级计算机中心，《操作指南：生成式人工智能的 GPU 内存使用量估算与性能分析》。 [在线]。链接：https://www.osc.edu/resources/getting_started/howto/howto_estimating_and_profiling_gpu_memory_usage_for_generative_ai [5] E. Erdil 和 D. Schneider-Joseph，《前沿模型训练中的数据移动限制》，arXiv 预印本 arXiv:2411.01137，2024年。[6] J. Zhao、Q. Lu、W. Jia、B. Wan、L. Zuo、J. Feng、J. Jiang、Y. Chen、S. Cao、J. He、K. Jiang、Y. Hu、S. Nong、Y. Peng、H. Lin、X. Liu 和 C. Wu，《OVERLORD：面向多源大型基础模型训练的数据加载器终极扩展》，arXiv 预印本 arXiv:2504.09844，2025年。[7] T. Singh，《如何使用 DeepSpeed 和 HuggingFace 在单个 GPU 上训练十亿参数自然语言处理模型》，JarvisLabs.ai 文档，2022年2月。[在线]。链接：https://docs.jarvislabs.ai/blog/ deepspeed-huggingface-training

[8]K. Sevegnani 和 G. Fiameni，《面向大语言模型训练的 NVIDIA Grace Hopper 高级优化策略》，*NVIDIA* 技术博客，2025年5月。 [在线]。链接：https://developer.nvidia.com/blog/advanced-optimization-strategies-for-llm-training-on-nvidia-grace-hopper/ [9] S. Zheng、W. Bao、Q. Hou、X. Zheng、J. Fang、C. Huang、T. Li、H. Du-anmu、R. Chen、R. Xu、Y. Guo、N. Zheng、Z. Jiang、X. Di、D. Wang、J. Ye、H. Lin、L.-W. Chang、L. Lu、Y. Liang、J. Zhai 和 X. Liu，《Triton-distributed：使用 Triton 编译器在分布式人工智能系统上编写可重叠执行的内核》，*arXiv* 预印本 *arXiv:2504.19442*，2025年。[10] A. Gangidi、R. Miao、S. Zheng、S. J. Bondu、G. Goes、H. Morsy、R. Puri、M. Riftadi、A. J. Shetty、J. Yang、S. Zhang、M. J. Fernandez、S. Gandham 和 H. Zeng，《面向 Meta 规模分布式人工智能训练的以太网上的 RDMA》，载于*ACM SIGCOMM* 会议论文集，2024年。[11] maknee，《云端大语言模型训练的网络与存储基准测试》，2025年。 [在线]。链接：https://maknee.github.io/blog/2025/ Network-And-Storage-Training-Skypilot/ [12] M. Kurzynski、S. Aga 和 D. Wu，《Chopper：一种多级 GPU 特征分析工具及其对大语言模型训练低效的洞见》，*arXiv* 预印本 *arXiv:2512.08242*，2025年。[13] J. Wang、J. Wang、J. Gao、Y. Cao、Y. Zhai、H. Wang、W. Song、D. Kong、L. Liu、W. Li 和 Z. He，《UniOrch：用于异构人工智能芯片高效大语言模型训练的统一混合框架》，*TechRxiv*。 [在线]。链接：https://www.techrxiv.org/users/972479/articles/1340215