低资源集群中的大语言模型分布式推理技术

Accelerating Distributed Inference of Large Language Models in Low-Resource Clusters

冯文佼/FENG Wenjiao， 李宗航/LI Zonghang，
虞红芳/YU Hongfang
（电子科技大学，中国 成都 611731）
（University of Electronic Science and Technology of China，Chengdu 611731，China）

DOI：10.12142/ZTETJ.202402007
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240404.2315.002.html
网络出版日期：2024-04-08
收稿日期：2024-02-20

摘要：探索了一种并行能力更强、具有更好兼容性的大语言模型（LLM）分布式推理范式。该范式专为弱算力、小显存环境设计。同时面向主机内外差异带宽，设计了基于通信树的高效All-Reduce组通信技术；针对小显存集群，设计了细粒度的显存管理与调度技术。最后，基于这些关键技术，构建了一套针对资源受限场景的LLM推理软件系统，旨在用数量有限的低资源设备，最大化能推理的LLM，同时通过优化通信策略与计算调度加速分布式推理。实验证明，在应用上述技术后，本方案的首词元生成延迟降低34%~61%，每秒生成词元吞吐量提升52%~150%，显存占用降低61%。

关键词：LLM分布式推理范式；资源受限场景；优化通信策略与计算调度

Abstract: A distributed inference paradigm for large language model (LLM) with stronger parallelism and better compatibility is explored, which is designed for weak computing power and small memory environments. Meanwhile, an efficient All-Reduce group communication technique based on communication tree is designed for the different bandwidths inside and outside the host, and a fine-grained memory management and scheduling technique is designed for small memory clusters. Finally, based on these key techniques, a set of LLM inference software system for resource-constrained scenarios is constructed, aiming to maximize the LLMs that can be inferenced with a limited number of low-resource devices, and at the same time accelerating the distributed inference by optimizing the communication strategy and computation scheduling. Experiments demonstrate that after applying the above techniques, the first lexical element generation latency is reduced by 34%~61%, the lexical element generation throughput per second is increased by 52%~150%, and the memory occupation is reduced by 61%.

Keywords: LLM distributed inference paradigm; resource-constrained scenarios; communication and computation scheduling optimization

引用格式：冯文佼, 李宗航, 虞红芳 . 低资源集群中的大语言模型分布式推理技术 [J]. 中兴通讯技术, 2024, 30(2): 43-49. DOI: 10.12142/ZTETJ.202402007
Citation: FENG W J, LI Z H, YU H F. Accelerating distributed inference of large language models in low-resource clusters [J]. ZTE technology journal, 2024, 30(2): 43-49. DOI: 10.12142/ZTETJ.202402007

作为科技革命的核心，人工智能（AI）在计算机视觉和自然语言处理等领域取得了重大进步。OpenAI在2022年底发布的ChatGPT[1]引领了大语言模型（LLM）时代，引发了对人工智能技术潜力的广泛探讨。然而，在全球AI技术竞争日益激烈和国际环境变化的背景下，高性能计算资源变得更加珍贵，尤其是面对AI芯片出口的限制，中国AI技术的独立发展变得迫在眉睫。由于存在技术鸿沟，中国AI硬件在短期内仍然面临着诸如弱算力、小显存和多机低互联带宽的技术挑战。为推动大模型AI产业的发展，中国学术界和工业界提出了在资源受限环境下进行LLM推理的策略。通过整合中低端算力资源，该策略实现超大模型的高效运行，既减少了对国外高端硬件的依赖，也为中小企业和教育机构提供了低成本的推理与部署方案，促进了国产AI计算卡的快速发展。因此，研究低资源环境下的LLM推理优化技术，成为了推动中国AI发展“降本增效”的关键。

现有LLM推理系统，如DeepSpeed[2]和FasterTransformer[3]，主要为强算力、高带宽、大显存的高性能智算中心提供高效的LLM推理能力。但与高性能智算中心相比，

在低资源条件下进行LLM推理仍存在一些不足。1）单张计算卡在算力和显存容量上面临明显限制。例如，NVIDIA的A100和H100这类图形处理器（GPU），在FP16运算性能上比中国的寒武纪思元、燧原邃思领先逾7倍，显存容量超5倍。对于中国的LLM推理应用来说，计算效率和存储能力成为了明显的瓶颈。2）多主机间的通信带宽远小于主机内的高速网络带宽。较大的模型不适合单张计算卡，需要依赖多卡服务器集群以适应显存。这也使我们能够将上述的计算成本和显存分摊到所有计算卡上，但代价是引入计算卡间通信。而在中低端数据中心内，主机间的网络带宽普遍限制在1~25 Gbit/s，与主机内可达100 Gbit/s的显存带宽和互联带宽相距甚远。这使得多机间互联网络的通信效率成为制约分布式推理性能的主要瓶颈。

此外，当前LLM主要采用Transformer架构[4]。它的主要思想是通过自注意力机制获取序列的全局信息，并将这些信息通过网络层进行传递。区别于传统的卷积神经网络（CNN）和循环神经网络（RNN），Transformer架构由于具有多个独立的注意力头，因此不需要按照时间步骤进行计算，具有更强的并行计算能力。为了实现最佳的性能和资源利用率，现在很多研究致力于自动混合并行推理，包括AlpaServe[5]、FlexFlow-Serve[6]和SpotServe[7]等。这些框架能够将自动搜索算法应用于LLM的推理过程，以确定最有效的并行策略。然而，分布式推理面临的主要挑战之一是数据通信产生了额外负担，因为这可能增加总体推理响应时间。尽管现有策略优化了并行计算，但它们往往忽略了针对Transformer架构特有通信需求的优化，这可能导致在推理过程中出现更加明显的延迟。

考虑到Transformer架构固有的内存密集型特性，高效的显存管理仍然是LLM分布式推理中面临的首要挑战。ZeRO-Offload[8]和ZeRO-Infinity[9]支持内存卸载，将GPU的显存压力分担到CPU甚至NVMe内存上，从而打破GPU的显存限制。但此类方法需要所有计算卡间拥有高速连接，因此使用场景将会受到很大的限制。

针对上述挑战，本文提出了适用于低资源集群的LLM分布式推理技术，实现用数量有限的低资源设备，最大化能推理的LLM，同时通过优化通信策略与计算调度来加速推理。

1 问题与动机分析

由于LLM推理对设备算力和显存容量有较高要求，Megatron-LM[10]通过张量并行将模型层，例如注意力、全连接前馈网络（FFN），从内部维度（例如头部、隐藏层）分割成多个部分，并将每个层部署在单独的计算卡上。但这种朴素的张量并行存在一个问题：自注意力的输出必须通过LayerNorm才能输入到FFN中进行计算。LayerNorm的正确性依赖于所有计算卡的自注意力结果，这是因为单卡结果无法确保其准确性。为此，Megatron-LM提出Reduce+LayerNorm+Broadcast算子，即计算卡完成自注意力输出后，先聚合（Reduce）到一卡执行LayerNorm，再将结果广播（Broadcast）回各卡继续多层感知机（MLP）计算。虽然该算子解决了LayerNorm层的并行问题，但它仍依赖单卡执行Reduce、LayerNorm及Broadcast。一方面，这种中心化的计算与通信算子会遭遇单点瓶颈；另一方面，这种算子的适用通信原语局限于Reduce和Broadcast，与诸多经典的All-Reduce通信库及其高效的All-Reduce原语实现（如Ring、Three-Phase Ring等）均不兼容。

由于典型数据中心的分层网络结构限制了跨主机带宽，All-Reduce的性能也会受阻。大规模LLM推理需要多主机合作以满足算力和显存要求。尽管单机多卡间可通过NVLink和高速串行计算机扩展总线标准（PCIe）实现高速通信，但各主机通常按机架分组并连接至架顶式（ToR）交换机。其中，机架内各主机通过1~25 Gbit/s的完整链路平分带宽进行互联，这限制了多主机间All-Reduce的通信效率。相关研究集中于优化模型训练阶段的All-Reduce通信，通过探测网络结构并制定分层聚合策略以适应网络变化，从而解决长期通信不平衡问题，但这并不完全适用于推理阶段。与训练不同，推理尤其是在线推理的持续时间较短，其核心目标是实现低延迟和高吞吐。因此，推理阶段更需针对带宽差异引致的通信瓶颈进行优化。

此外，为了实现用数量有限的低资源设备最大化能推理的LLM，同时考虑到Transformer架构固有内存密集性，高效的显存管理仍然是LLM分布式推理中面临的首要挑战。现有推理系统[3-11]基于高性能智算中心开发了一系列内存卸载技术，例如：通过频繁通信实现了GPU显存负载转移至CPU或NVMe存储，有效突破显存限制。然而，这些推理系统往往沿用了为训练阶段设计的卸载技术[8-9,12-14]，直接应用于资源受限的分布式推理可能不理想。因为这些技术在资源受限环境下可能导致对更多计算卡和高并行度的依赖，增加通信复杂性，并且主机间的低带宽难以支持这种强度的通信。同时，这些技术忽略了生成推理的特殊计算属性，未能利用面向吞吐量的LLM推理计算的结构，并错过了有效调度输入输出（I/O）流量的绝佳机会。这些先前的工作促使我们设计一套适用于低资源集群的LLM分布式推理技术方案。该方案引入了一种高兼容的分布式推理范式，同时特别

关注主机内外带宽差异以及如何最大化LLM推理的潜力。

在本文中，我们研究了一种面向弱算力、小显存的高兼容的分布式推理范式。该范式能支持All-Reduce通信原语。具体而言，我们揭示了在进入非线性层之前，LayerNorm和Broadcast两个操作是可交换的。基于此我们提出了一个创新方案：将传统的Reduce+LayerNorm+Broadcast算子简化为All-Reduce+LayerNorm算子。这一新范式旨在全面支持All-Reduce通信原语，使之能在不同场景中利用多样化的通信库来实现高效的分布式推理。

在中低端数据中心内机架规模下涉及跨主机的All-Reduce通信时，分布式推理低带宽网络会带来明显的性能瓶颈问题。为解决这一问题，我们提出了一种面向主机内外差异带宽的高性能All-Reduce通信算法。具体来说，我们根据主机内和机架内带宽特点的差异性，实现基于通信树的高效All-Reduce组通信库，有效组织分布式推理的中间计算结果聚合与分发，从而减少跨主机通信并充分利用内部高速带宽。

此外，我们还探索了面向LLM、小显存集群的显存管理与调度，旨在低资源环境中实现更大规模与更高效的LLM推理。我们采用了动态调度模型参数的方法，包括及时回收未使用的参数空间以减少显存占用，并预加载即将使用的参数以消除轮次间的等待时间，从而无缝加速推理过程。这种策略通过细粒度控制显存的使用，降低了峰值显存需求，即使在显存有限的硬件条件下也能高效地执行大规模模型推理，在确保推理性能的同时提高硬件资源的使用效率和成本效益。

为了实现上述想法，一些技术难题仍需要解决：

难题1：如何保证本文提出的范式在保持理论计算正确性的同时，与现有张量并行范式具有等价的推理计算效率和资源消耗。

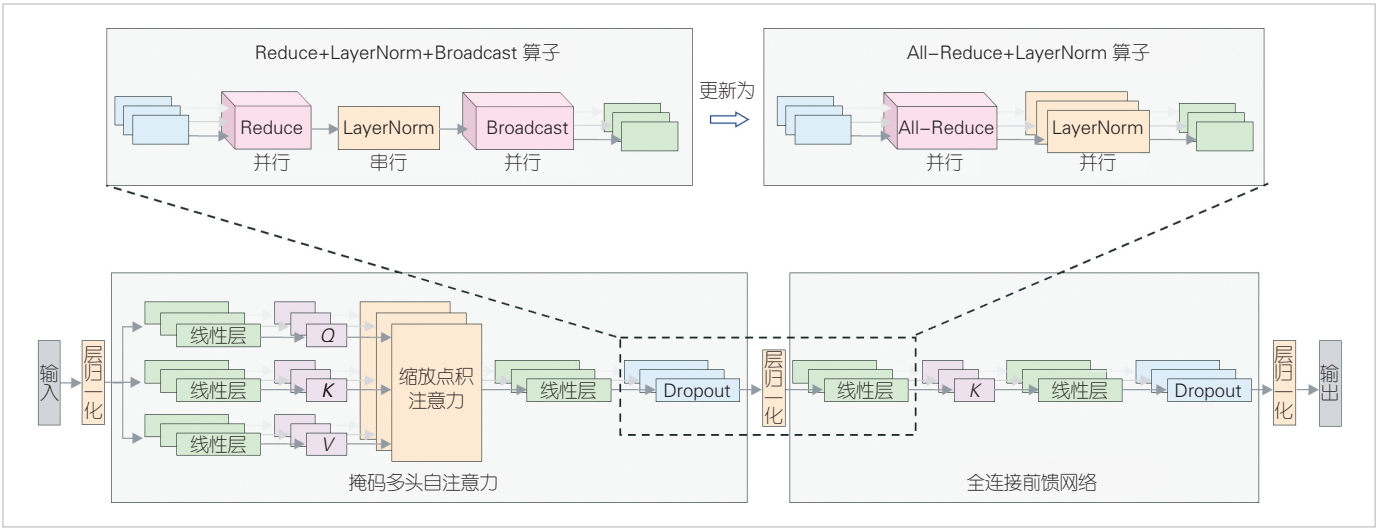
难题2：如何保证本文提出的分层聚合算法在理论计算结果正确的前提下，最大限度地减少由机架内带宽的低带宽网络引起的通信开销。

难题3：基于新型分布式训练范式，如何精准把控推理过程中所需的模型参数在显存中的加载和卸载时机，即明确何时将这些相关数据载入显存以进行有效的计算，以及何时将其从显存中移除以优化资源使用。

2 方案设计

2.1 面向弱算力、小显存的高兼容的分布式推理范式

Megatron-LM采用张量并行来应对大规模模型对高算力和大显存的强依赖，并通过引入Reduce+LayerNorm+Broadcast算子确保计算的准确性。然而，这种中心化的并行范式存在单点瓶颈，并且不支持如信息传递接口（MPI）、NVIDIA集合通信库（NCCL）等主流All-Reduce通信库，影响了其兼容性和效率。为此，我们研究了一种兼容性更好的张量并行范式。该范式能支持All-Reduce通信原语，使之能在不同场景中利用多样化的通信库来实现高效的分布式推理，并且与现有张量并行范式一样具有等价的推理计算效率和显存消耗。具体如图1所示，在进入MLP之前LayerNorm和Broadcast两个操作是可交换的。基于这一关键发现，本文提出将Reduce+LayerNorm+Broadcast算子合并为All-Reduce+LayerNorm算子。接下来，我们将从理论正确性和资



▲图1 面向弱算力、小显存的高兼容分布式推理范式示意图

源消耗两个维度来深入分析这一新算子的性能。

在计算结果理论正确性方面，该推理范式先在单个计算卡上对Reduce后数据进行LayerNorm，再将结果Broadcast给其他计算卡。这与先将Reduce后的结果Broadcast给其他计算卡，再在各计算卡上分别进行LayerNorm，计算得到的结果等价。而在资源消耗方面，All-Reduce+LayerNorm算子不会牺牲显存，因为中间结果Z通过各计算卡并行LayerNorm操作。得到Z之后，LayerNorm产生的临时变量立即被释放。因此，这种方法主要影响峰值显存使用而非总显存。

总的来说，我们提出一种高兼容的分布式推理范式，该范式将Reduce+LayerNorm+Broadcast算子合并为All-Reduce+LayerNorm算子，在确保计算结果理论正确性、资源消耗等价的前提下，统一LLM张量并行范式的通信原语为All-Reduce。一方面，我们可以根据实际环境，灵活使用MPI、Gloo、NCCL、Hovorod、PS等第三方通信库（或自研通信库）来满足个性化的推理需求；另一方面，All-Reduce原语的执行效率比分别执行Reduce和Broadcast原语更高，具有更宽阔的通信优化空间。

2.2 面向主机内外差异带宽的高性能All-Reduce通信算法

当前基于All-Reduce的通信优化研究主要集中在缓解模型训练阶段的长期通信不平衡问题。然而，在追求低延迟和高吞吐的推理阶段，跨主机All-Reduce通信中低带宽网络引发的瓶颈问题更值得关注。如图2所示，数据中心的分层拓扑结构将机器分至机架并连接至ToR交换机，以保障机架内主机间共享完整链路带宽。但带宽通常局限于1~25 Gbit/s，这与主机内多卡间上百Gbit/s的高速通信相距甚远[15]。为此我们开发了一种基于通信树的高效All-Reduce组通信库，以减少跨主机通信并充分利用内部高速带宽。具体来说，对于一次推理任务，当需要进行All-Reduce操作时，通信树的构造过程如下：

首先在各主机内部选出性能最优的计算节点作为本地主节点（LM），用于本地聚合。所有主机中的LM之一被选为全局聚合的全局主控（GM）。通信树的通信按照以下4个步骤完成：

1）每个计算节点将自己的局部计算结果发送到各自的LM（仅限主机内推理流量）；

2）所有的LM将本地聚合结果发送至GM以进行全局聚合（仅主机间流量）；

3）GM完成全局聚合，然后使用反向路由将全局聚合结果传播回LM；

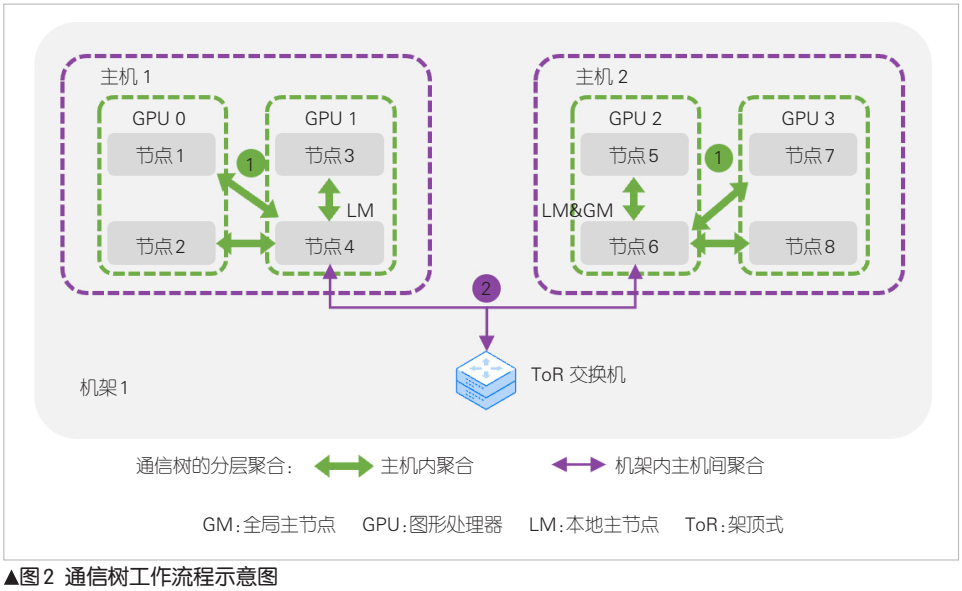
4）LM将全局聚合块扇出到所有计算节点。

与朴素All-Reduce主机间通信次数相比，采用通信树策略后，主机间的All-Reduce通信次数降为仅需主机的数量-1次。可以看到，本方案充分利用了数据中心网络的结构特点，通过优化内部节点的通信路径，有效管理推理中间结果的聚合和分发过程，从而大幅度降低了主机间的通信频率和通信延迟，提高了整体推理效率。

2.3 面向LLM、小显存集群的显存管理与调度

尽管针对LLM的内存卸载技术在高性能计算中心依然有效，但这些主要为训练而设计的技术在资源受限的分布式推理场景中应用可能不理想。它们可能增加对计算资源的依赖，加重通信负担，同时忽略了推理特有的计算需求和优化吞吐量的机会。通常来讲，LLM推理包含预填充和解码两阶段。其中，预填充阶段并行处理输入，解码阶段依赖之前所有tokens信息生成新tokens。为提高效率，现有工作提出将这些信息以键（K）和值（V）的形式缓存于显存中，大大减少了重复计算次数。但随着对长序列推理的需求不断增长，与模型权重和其他激活所需的工作空间相比，KV缓存的显存占用成为主要优化目标。

为解决这个问题，我们开发了一套针对LLM和小显存集群的细粒度显存管理与调度机制，目



▲图2 通信树工作流程示意图

的是在资源受限的环境下，实现更大规模的LLM推理。其中，这一机制包含两个核心模块：最小化显存占用机制和预加载机制。通过将Transformer模型的每个Layer视为独立状态，并将参数分散到不同GPU上，最小化显存占用机制确保每个计算单元仅保留当前必需的参数片段，大幅降低了总体显存需求。同时，预加载机制能在当前计算进行前加载下一步所需的参数，有效消除了推理过程中的等待时间，进一步提升了推理效率。这两个模块的协同工作，使得我们的显存管理策略能够在减少资源消耗的同时保证模型推理的连续性和吞吐。

1）最小化显存占用机制

已知Transformer模型由多个Layer串连而成，我们将每个Layer视为一个独立状态，同时，根据新型分布式推理范式将每个Layer中的参数切分为不同的部分。每个GPU仅维护对应的部分。对于特定的推理计算，用b表示批量大小，s表示输入序列长度，n表示输出序列长度，h表示隐藏维度，L表示Transformer层数，a表示attention heads。考虑有N个GPU执行推理。对于batch X，GPU_n处理Layer 1到Layer L的s_{n1}~s_{n2}参数片段，包括a_{n1}~a_{n2}注意力头和相应的MLP参数切片。针对batch X中的某个prompt生成一个词元过程，具体的显存管理与调度流程如图3所示。

显存管理与计算线程并行运行，前者负责模型参数在显存与内存间的调度，后者执行GPU上的张量并行计算。在推理的每个步骤i中，系统识别Layer(i)作为当前的活动状态。对于GPU_n，其计算线程专注于执行Layer(i)内部特定的s_1~s_2参数切片的计算任务。与此同时，显存管理线程负责从显存中卸载掉之前步骤Layer(i − 1)的s_1~s_2参数切片和对应注意力头的KV缓存。与传统方法相比，显存需完整存储模型参数及KV缓存。本策略确保计算卡在任一时刻仅保留必要的参数，从而在资源受限的环境中实现更大规模的模型推理。

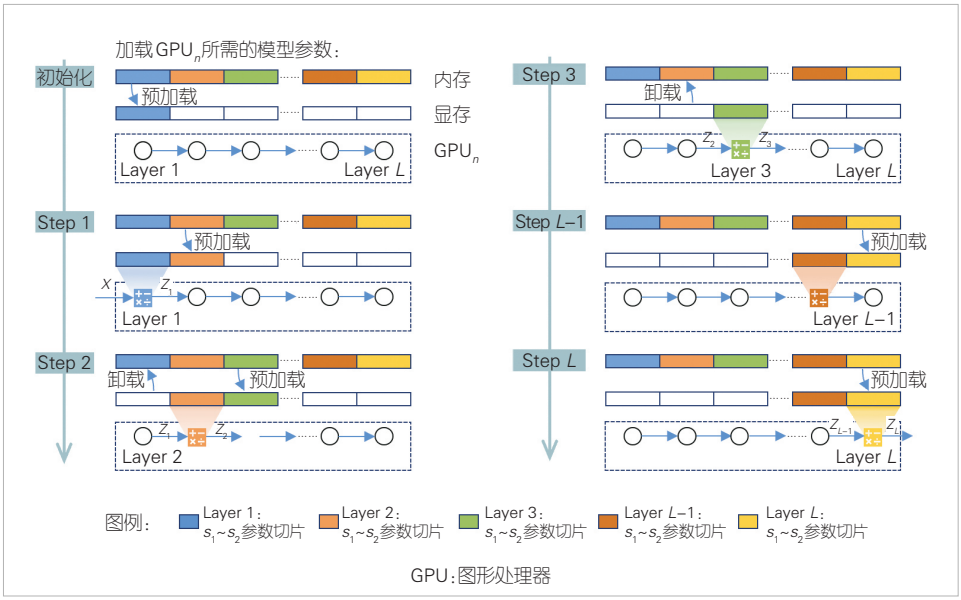
2）预加载机制

然而，上述的串行计算存在一个明显的缺陷：在完成Layer(i)的计算后，推理过程需等待Layer(i + 1)的参数加载至显存，这显著降低了推理效率。因此，我们引入了预加载机制，允许显存管理线程在Layer(i)计算进行时，提前将Layer(i + 1)的s_1~s_2参数切片和对应注意力头的KV缓存从内存预加载至显存。该机制使用少量的显存，消除了计算停滞，保障了推理流程的无缝衔接。

3）支撑的模型范围对比分析

在资源受限条件下，通过上述细粒度的显存管理与调度，这套范式理论上可以支撑多大的模型？考虑fp16中的GPT3-175B模型（L=96，h=12 288），峰值时存储KV缓存的总字节数为$4 \times b \times h \times (s+n)$。模型所需的总显存为350 GB，存储KV缓存所需的总显存为816 GB（其中，b=16，s=512，n=32），总显存需求约为1 166 GB，平均每层需要12 GB显存。在四卡系统中，每卡理论上承担3 GB，即使考虑额外的缓冲和预加载空间，每卡的显存使用也不会超过6 GB。相比之下，传统GPU-only方案每卡需承担290 GB。因此，我们的方法可以推理比GPU-only的解决方案大48倍的模型（每卡承担6 GB与290 GB），超越DeepSpeed Inference的25倍[3]。这表明我们所提方法在模型扩展性方面具有优越性。

总的来说，此方法在保证推理性能的基础上，通过细粒度管理与调度显存，在显存受限的硬件环境中也能高效推理更大规模的模型，极大提高了硬件资源的利用率和成本效益。



▲图3 细粒度的显存管理与调度流程图

3 实验评估

本节我们将围绕两个方面来评估所提方案的性能：1）对首词元生成延迟的降低以及每秒生成词元吞吐量的提升效果；2）量化本系统显存管理与调度机制的优势。

3.1 实验平台设置

在构建系统时，我们选用

PyTorch[16]作为核心框架。在计算方面，我们重新设计了分布式推理的架构，并实现了精细的显存管理及调度策略。在通信层面上，我们依托PyTorch-DDP，打造了一种基于通信树的高效All-Reduce集群通信机制。我们在两台配置有双Intel(R) Xeon(R) E5-2678 v3 CPU、4块NVIDIA RTX 2080TI GPU、128 GB系统内存及44 GB总显存的主机上开展实验。主机间通过1 GB网络带宽互联。实验采用Meta AI发布的LLaMA-3B。表2展示了默认的超参数配置。

3.2 延迟和吞吐量

我们选择基于分布式数据并行（DDP）的原生All-Reduce作为Benchmark，采用参数服务器（PS）架构。worker节点通过采用“星形”拓扑结构进行通信，即多个worker直接与中心服务器进行数据交换。

我们首先对Benchmark和本方案在首词元生成延迟及每秒生成词元吞吐量方面进行了比较测试。其中，首词元生成延迟涵盖模型处理输入并自回归生成下一词元的计算及通信延迟，每秒生成词元吞吐量用每秒可以处理的词元数来衡量。如图4所示，我们测试了不同的输入词元数。相比于Benchmark，本方案的首词元生成延迟降低34%~61%，每秒生成词元吞吐量提升52%~150%。这证明了上述技术的有效性。

▼表2 LLaMA-3B模型默认超参数设置

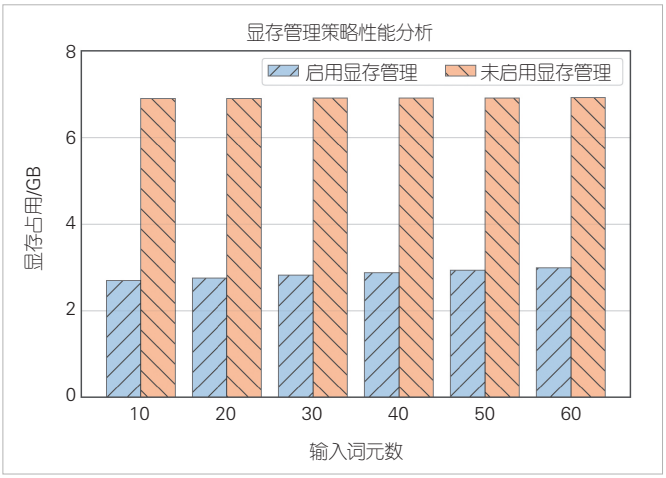
变量名	符号	值
注意力机制中的头数	N_{atten_heads}	32
批量大小	B_{size}	32
隐藏层的维度大小	H_{model}	3 200
模型中的层数	N_{layers}	26
序列长度	seq_len	2 048
词汇表的大小	vocab_size	32 000

3.3 显存占用

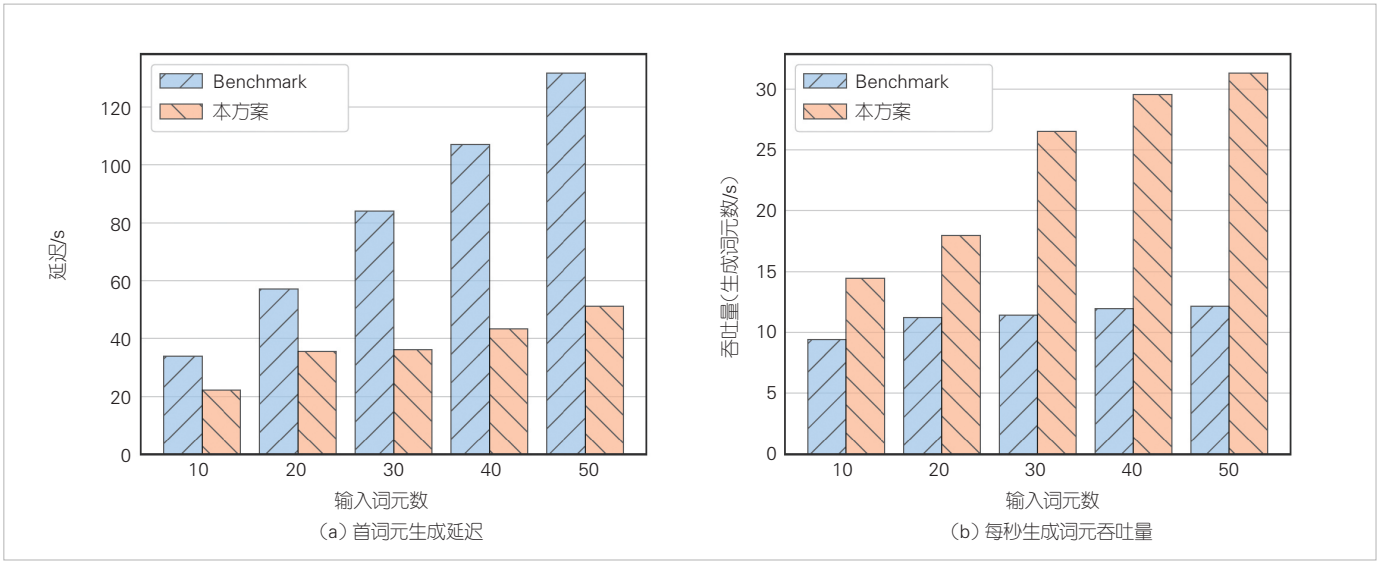
为评估本方案的显存管理与调度效能，我们比较了启用与未启用本显存管理方案时，首词元生成阶段节点的峰值显存占用情况。图5展示当输入词元数量增加时节点显存占用的线性增长趋势。与未启用显存管理相比，本方案的显存占用降低61%。这也验证了2.3节中的分析。

4 结束语

在面对全球竞争和资源限制的挑战下，我们提出了一种适应弱算力及小显存环境的分布式LLM推理架构。同时通



▲图5 显存管理对节点显存占用的影响



▲图4 不同方案下延迟和吞吐量对比

过独创的适应性通信策略和显存管理方案，我们有效克服了带宽和显存限制，构建了一个高效推理框架，使得有限资源下的LLM推理成为可能。此项成果推进了中国AI的自主发展，为中国AI产业的发展和全球技术多样性贡献了重要力量。

致谢

感谢电子科技大学信息与通信工程学院赵舒心和熊彦旭硕士对本研究技术与实验部分的贡献！

参考文献

[1] OpenAI. ChatGPT [EB/OL]. (2022-12-30) [2024-02-25]. https://openai.com/blog/chatgpt
[2] AMINABADI R Y, RAJBHANDARI S, AHMAD AWAN A, et al. DeepSpeed-inference: enabling efficient inference of transformer models at unprecedented scale [C]//Proceedings of SC22: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2022: 1-15. DOI: 10.1109/SC41404.2022.00051
[3] NVIDIA. FasterTransformer [EB/OL]. (2022-03-20)[2024-02-25]. https://github.com/NVIDIA/FasterTransformer
[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. ACM, 2017: 6000 - 6010. DOI: 10.5555/3295222.3295349
[5] LI Z H, ZHENG L M, ZHONG Y M, et al. AlpaServe: statistical multiplexing with model parallelism for deep learning serving [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2302.11665
[6] Github. FlexFlow [EB/OL]. [2024-02-25]. https://github. com/Flexflow/FlexFlow/tree/inference
[7] MIAO X P, SHI C N, DUAN J F, et al. SpotServe: serving generative large language models on preemptible instances [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2311.15566
[8] REN J, RAJBHANDARI S, AMINABADI R Y, et al. ZeRO-offload: democratizing billion-scale model training [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2101.06840
[9] RAJBHANDARI S, RUWASE O, RASLEY J, et al. ZeRO-infinity: breaking the GPU memory wall for extreme scale deep learning [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2104.07857
[10] SHOEYBI M, PATWARY M, PURI R, et al. Megatron-LM: training multi-billion parameter language models using model parallelism [EB/OL]. [2024-02-25]. http://arxiv. org/abs/1909.08053.pdf
[11] HuggingFace. Hugging face accelerate [EB/OL]. [2024-02-25]. https://huggingface.co/docs/accelerate/index
[12] LI Y J, PHANISHAYEE A, MURRAY D, et al. Harmony: overcoming the hurdles of GPU memory capacity to train massive DNN models on commodity servers [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2202.01306
[13] HUANG C C, JIN G, LI J Y. SwapAdvisor: pushing deep learning beyond the GPU memory limit via smart swapping [C]// Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2020: 1341 - 1355. DOI: 10.1145/3373376.3378530
[14] WANG L N, YE J M, ZHAO Y Y, et al. Superneurons: dynamic GPU memory management for training deep neural networks [C]//Proceedings of the 23rd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. ACM, 2018: 41 - 53. DOI: 10.1145/3178487.3178491
[15] LUO L, NELSON J, CEZE L, et al. Parameter hub: a rack-scale parameter server for distributed deep neural network training [C]//Proceedings of the ACM Symposium on Cloud Computing. ACM, 2018: 41-54. DOI: 10.1145/3267809.3267840
[16] PASZKE A, GROSS S, MASSA F, et al. Pytorch: an imperative style, high-performance deep learning library [EB/OL]. (2019-12-03)[2024-02-25]. https://arxiv.org/abs/1912.01703

作 者 简 介

冯文佼，电子科技大学在读硕士研究生；研究方向为分布式机器学习系统及其优化技术、大模型分布式推理优化技术。

李宗航，电子科技大学在读博士研究生、牛津大学和南洋理工大学访问学者；研究方向包括分布式人工智能、联邦学习和大模型分布式计算；相关研究入选中国通信学会2021领先创新科技成果；发表论文20余篇，授权中国发明专利6项，出版学术著作1部。

虞红芳，电子科技大学教授、博士生导师，信息与通信工程学院副院长；长期致力于智慧网络及应用研究；受邀在全球学术会议上做报告10余次，担任3个网络领域全球高水平期刊的副主编；获得2016年教育部自然科学奖二等奖，主持研发的“跨数据中心高性能分布式机器学习系统GeoMX”和“基于轻量级虚拟化的大规模网络创新平台Klonet”分别获中国通信学会2021年未来网络领先创新科技成果奖、2021年网络5.0创新科技成果奖；发表论文100余篇；授权中国发明专利30余项、美国发明专利2项，出版学术专著4本。