doi:10.13756/j.gtxyj.2026.260135.
专题:智算中心光网络算网融合关键技术
杨帅,闫付龙,杨鸿瑞,等.十万卡智算网络流量建模与加速仿真研究[J].光通信研究,2026(4):10-17.
Yang S,Yan F L,Yang H R,et al. Research on Traffic Modeling and Accelerated Simulation for Hundred-Thousand-GPU AI Computing Networks[J]. Study on Optical Communications,2026(4):10-17.

十万卡智算网络流量建模与加速仿真研究(特邀)

杨 帅,闫付龙,杨鸿瑞,薛植文,连博宇,张 杰
(北京邮电大学 信息光子学与光通信国家重点实验室,北京 100876)

摘要:【目的】随着人工智能(AI)的快速发展,大模型的参数规模不断扩大,训练和推理环节对算力的需求不断增加,推动了超大规模智算网络的建设。智算网络节点间的通信开销是限制算力发挥的主要瓶颈,因此在建设前期,使用离散事件仿真器对网络性能进行评估,对于优化网络设计和降低节点间通信开销具有重要意义。但智算网络通信流量需求巨大且具有时空依赖关系,传统全量仿真面临巨大状态开销,网络拓展性与仿真效率之间形成矛盾。【方法】文章首先分析了大模型分布式训练场景下的通信特征,将 DeepSeek 架构的稠密与稀疏两类模型训练时的集合通信原语,建模为具有时空依赖关系的点对点流量,并对流量特征进行了分析。随后,基于所得流量的周期性和大模型训练时集群网络的收敛性,提出了稳态外推加速算法:利用 OMNeT++平台捕捉仿真过程中网络的瞬态变化,在仿真初期,对通信流量进行包级仿真;在网络进入稳态后,将离散的包级仿真抽象为流级,依据最大最小公平原则解析流速率,对剩余流量的仿真过程进行数学外推。【结果】在 3 种网络规模上,对上述仿真加速效果进行验证,实验结果表明,该方法在保留网络底层拥塞特征的同时,总仿真时间实现了最高 4.7 倍的加速,并使万卡规模仿真时的内存占用量降低了 30%;随后,在十万卡规模的网络上进行了小规模验证,在仿真所得网络特性接近基准的前提下,将仿真时长降低了 73%,并使内存占用量降低了 31%。【结论】文章将集合通信建模为细粒度的点对点流量,为智算网络的评估提供了数据支撑。文章所提稳态外推加速算法有效提升了大规模智算网络的仿真效率,为更大规模智算网络的性能评估提供了参考。

关键词:智算网络;流量建模;集合通信;离散事件仿真
中图分类号:TN929 **文献标志码**:A

Research on Traffic Modeling and Accelerated Simulation for Hundred-Thousand-GPU AI Computing Networks

YANG Shuai, YAN Fulong, YANG Hongrui, XUE Zhiwen, LIAN Boyu, ZHANG Jie
(State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract:【Objective】As Artificial Intelligence (AI) accelerates, the parameter scale of foundation models continues to expand. The demand for computing resources in training and inference links continues to increase, catalyzing the deployment of massive AI computing infrastructures. In intelligent compute fabrics, inter-node communication overhead is the chief bottleneck stifling compute utilization. Consequently, profiling network performance with discrete-event simulators before build-out is paramount for optimizing design and slashing these overheads. However, the amount of communication traffic is huge and has a time-space dependence. Traditional full simulation faces huge state overhead, and there is a contradiction between network scalability and simulation efficiency.【Methods】Firstly, we analyzed the communication characteristics in the distributed training scenario of large models. The set communication primitives of the dense and sparse models of the DeepSeek architecture are modeled as point-to-point traffic with spatio-temporal dependencies, and the traffic characteristics are analyzed. Then, based on the periodicity of the obtained traffic and the convergence of the cluster network during the training of the large model, a steady-state extrapolation acceleration algorithm is proposed. The OMNeT++ platform is used to capture the transient changes of the network during the simulation process, and the packet-level simulation of the communication traffic is carried out at the initial stage of the simulation. After the network enters the steady state, the discrete packet-level simulation is abstracted as a flow level. The flow rate is analyzed according to the principle of maximum and minimum fairness, and the simulation process of the remaining traffic is mathematically extrapolated.【Results】The above simulation acceleration effect is verified on three network scales. Experiments show that this method achieves a maximum of 4.7 times acceleration of the total simulation time while retaining the underlying congestion characteristics of the network. It can also reduce the memory usage of the card scale simulation by 30%. Subsequently, a small-scale verification was performed on a 100,000-card network. Under the premise that the network characteristics obtained by simulation are close to the benchmark, the simulation time is reduced by 73%, and the memory usage is reduced by 31%.【Conclusion】In this paper, collective communication is modeled as fine-grained point-to-point traffic, which provides data support for the evalua-

收稿日期:2026-05-14; 修回日期:2026-06-09; 纸质出版日期:2026-08-10
基金项目:国家自然科学基金青年资助项目(62301062)
作者简介:杨帅(2000-),男,河北承德人。硕士,主要研究方向为光网络。
通信作者:闫付龙,副教授。E-mail:f.yan@bupt.edu.cn

tion of intelligent computing networks. The proposed steady-state extrapolation acceleration algorithm effectively improves the simulation efficiency of large-scale intelligent computing networks and provides a reference for the performance evaluation of larger-scale intelligent computing networks.
Key words: AI computing network; traffic modeling; collective communication; discrete-event simulation

0　引　言

大模型分布式训练(Distributed Model Training, DMT)时,节点间通信流量具有复杂的时空依赖关系[1],且模型的训练迭代过程会产生海量重复的通信流量,使用离散事件仿真器对上述流量进行仿真,将产生巨大的状态开销[2]。当前国内外研究主要有两条路线:一是将网络仿真粒度由包聚合为更粗粒度的流,牺牲部分精度来降低仿真计算的复杂度,实现仿真加速。Li 等人提出了基于机器学习的流级仿真算法,该算法把网络状态的时空转移过程解耦,在实现加速的同时,大幅降低了单流估计误差[3];Wang 等人提出的 HyGra 架构,在流量突发期采用包级仿真,将平稳期切换到流级计算,有效提升了仿真速度[4]。二是不增大仿真粒度,通过优化任务调度,发挥硬件算力从而实现加速。Bai 等人提出了并行仿真内核 Unison,将任务细粒度划分,再采用自适应调度,动态分配任务到不同的线程,提高了多核处理器的利用率[5],加快了仿真速度;Gao 等人提出了面向数据设计的网络仿真(Distributed Data-Oriented Network Simulator, DONS)框架,该框架面向数据设计,主要思路是优化底层缓存,并采用和 Unison 类似的任务调度模式,实现任务的自动分配,以损失少量精度为代价,释放了多核处理器的并发算力[6]。在仿真超大规模集群时,上述思路均存在局限性:粗粒度仿真无法仿真网络的微观特征,多线程加速存在硬件门槛,随着仿真规模的扩大,加速效果呈现亚线性拓展(Sub-Linear Scaling)特征[7]。

总体来看,现有研究未充分利用 DMT 通信流量的周期性特征,本文的思路是:将大模型 DMT 通信建模为具有时空依赖关系的点对点(Peer-to-Peer, P2P)流,基于流量的周期性特征,提出了一种稳态外推加速算法。实验表明,该算法实现了最高 4.7 倍的仿真加速,并将仿真时内存占用量降低约 30%。有效降低了超大规模智算网络仿真的计算开销。

1　通信流量细粒度建模

随着大模型的参数规模向万亿级发展,单机单卡难以满足训练需求,为了解决算力增速不足的问题,DMT 技术成为必然选择。DMT 通常采用张量并行(Tensor Parallel, TP)、数据并行(Data Parallel, DP)、流水线并行(Pipeline Parallel, PP)[8]和专家并行(Expert Parallel, EP)[9]等策略,将模型或数据切分并分配到不同节点进行训练或推理,为了同步分片数据,这些节点间会产生海量的通信,降低单卡算力利用率。以 Megatron-LM 框架为例[10],当集群扩展至 3 072 个图形处理器(Graphics Processing Unit, GPU)时,即使在高度优化的系统中,单卡算力利用率仍降至约 52%。节点间的通信开销成为限制算力释放的主要因素[11]。集群组网方式是影响节点间通信开销的重要因素,如果基于实际训练时的通信反馈,对组网方式进行调优,会消耗海量算力。DeepSeek-V3 技术报告指出,其单次完整训练耗费约 278.8 万个 H800 GPU 小时[12],因此,在集群建设初期,借助仿真对网络性能进行评估具有重要价值。

在 DMT 中,不同的并行策略会在训练和推理阶段触发通信交互,图 1 所示为 N_0～N_3 4 个计算节点(记为 N_i, i=0, 1, 2, 3)进行集合通信时的状态变化过程。x_i 为节点 N_i 初始时携带的数据量,x_i 切分后的第 j 个分片数据量记为 $x_i^{(j)}$(j=0, 1, 2, 3,对应 4 个节点各自负责处理的数据分片)。通信操作主要包含 4 类:①全收集(All-Gather, AG)负责将各节点的数据分片聚合成完整的全局数据副本,如图 1 所示,各节点将自身的数据(如 N_0 的 x_0)发送至其他节点,同时接收来自其余节点的数据,所有节点通过拼接累积数据块完成全局同步;②全对全(All-to-All, A2A)主要出现于混合专家(Mixture of Experts, MoE)架构的特征路由(Token Routing)阶段,本质是跨节点的数据转置。如图 1 所示,节点 N_i 将本地数据划分为多个分块,并将数据块 $x^{(j)}$ 发送至节点 N_j;同时,N_i 也会从其他节点接收属于自身的数据块 $x^{(i)}$,最终完成按分块索引的聚合;③规约散布(Reduce-Scatter, RS)分为规约和散布两个阶段。规约阶段对不同节点上索引相同的数据执行并行规约,散布阶段将规约后的数据按节点索引分发[13],确保每个节点仅持有一块规约后的全局数据分片;④全归约(All-Reduce, AR)要求所有节点参与交换并计算全局梯度,这一过程可以看作是 RS

和 AG 的整合，如图 1 所示，节点 N_i 先通过对本地数据进行规约，得到分片数据总和并按索引分发，使得每个节点获得部分数据分片，接下来进行 AG 操作，将各节点所持有的局部结果同步至所有节点，最终使每个节点均获得完整的全局数据。

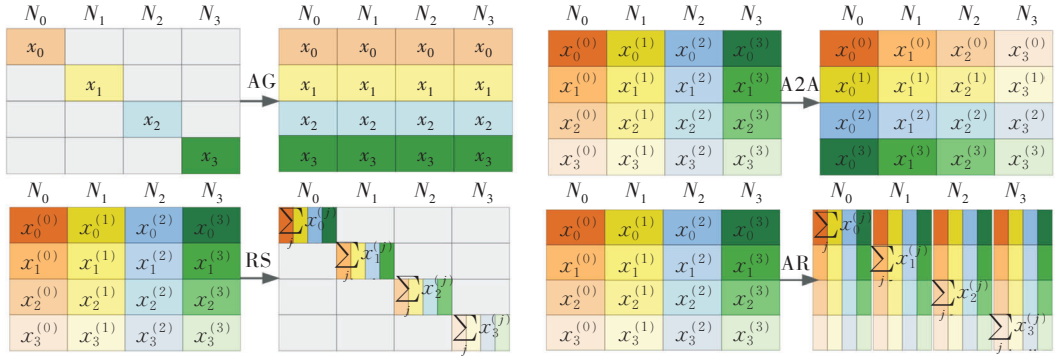
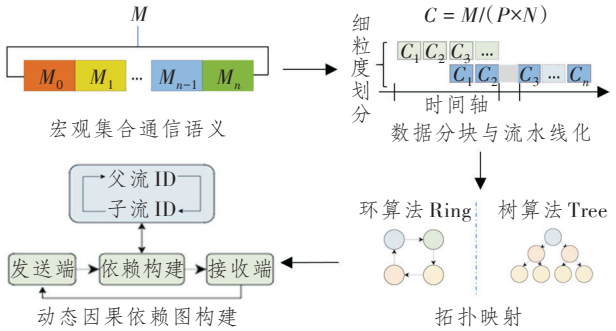


图 1　4 个节点参与集合通信的初态与终态

Figure 1　Initial and final states of 4 nodes in collective communication

传统网络仿真器通常将上述集合通信抽象为不可分割的单一宏观流量块[14]，而真实训练场景中，集合通信由一系列复杂的 P2P 流量交互组成。本文基于 NVIDIA 集合通信库（NVIDIA Collective Communications Library，NCCL）设计并实现了一个 P2P 流量解析框架，基于拓扑感知与细粒度数据分块，将宏观的集合通信原语建模为具有依赖关系的 P2P 执行流。关键步骤如图 2 所示。



注：M_n 为第 n 个节点携带的数据片；C 为微块大小，C_n 为随时间推进的第 n 个数据块；M 为同步的总数据量；P 为通信域内的 GPU 数量；N 为并发通道数。

图 2　微观流解析框架概述

Figure 2　Overview of the micro-flow analysis framework

针对大模型训练中复杂的跨域通信需求，流量建模时首先根据操作类型对通信域进行隔离。并行域在物理上的映射空间需满足以下约束：

$$N_{\mathrm{GPU}} = S_{\mathrm{TP}} \times S_{\mathrm{DP}} \times S_{\mathrm{PP}}, \tag{1}$$

式中：N_{GPU} 为参与训练的 GPU 总数；S_{TP} 为 TP 度；S_{DP} 为 DP 度；S_{PP} 为 PP 度。在此约束下，拆分后的 P2P 流量能映射至底层物理网络的特定子集，反映不同并行策略下的节点通信状态。明确空间路由映射后，为模拟 NCCL 多通道并发和步进块传输的微观行为，将总载荷切分为细粒度的微块，以环形集合通信为例，分块大小可定义为

$$C = \frac{M}{P \times N}。 \tag{2}$$

为保证 P2P 流量在时序上的因果性，框架采用双向指针构建流量间的依赖关系。在环形拓扑[15]中，每执行一次步进，指针会追踪上一轮次（记为 $r-1$ 轮）生成接收端流标识，并将其作为当前轮次（第 r 轮）发送的前提，确保数据流转时满足时序因果性；树形拓扑中[16]，则通过节点的计数器实现唤醒。树型拓扑的通信范式主要有两类：规约收集和广播分发。在规约阶段，数据自底向上传递，初始化时会为每个父节点设置一个入度计数器，初值为该节点所有子节点数量，每当子节点完成向上的流量传输后，父节点的计数器递减。当入度计数器归零时，父节点被激活，将接收的子节点流量标识作为自身流量的发送前提，以保持因果性。规约阶段，将父节点 v 向上发送数据的激活条件定义为

$$a_v = I\left(\sum_{u \in C_v} x_{u,v} = |G_v|\right), \tag{3}$$

式中：a_v 为节点 v 在规约阶段的激活状态，当 a_v=1 时，节点 v 被激活并向上游节点发送数据；G_v 为节点 v 直接子节点集合，$|G_v|$ 为子节点集合的大小，即初始入度；$x_{u,v}$ 为传输状态指示变量，子节点 u 至父节点 v 的数据被成功接收时取值为 1，否则取 0；$I(\cdot)$ 为指示函数，条件成立时取 1，否则取 0。

在广播分发阶段，数据自顶向下传递，每个节点有唯一的直接父节点，入度计数器初始化为 1，出度计数器初始化为直接子节点的总数。数据传输时，节点设定为等待状态，监听上游数据到达事件。当节点完整接收父节点的数据后，满足向下游发送数

据的时序条件，入度计数器归零，节点被激活。激活后，节点读取自身的出度计数器，若出度>0，则将接收到的数据作为源，向下游所有直接子节点并行发送数据。

为验证上述解析框架在真实场景下的通用性，本文选取 DeepSeek-LLM-7B 与 DeepSeek-R1-671B 模型[17-18]DMT 工作负载作为样本，进行建模分析。前者为标准的密集型架构，总参数量与激活参数量相等，单次前向与反向传播中所有参数参与计算，代表中小规模语言模型的计算特征；后者为 MoE 架构，总参数量远大于激活参数量，每次训练仅部分参数参与运算，对应超大规模模型典型的稀疏计算特征。表 1 所示为两类模型多维特征对比。

表 1　参数密集型与 MoE 模型多维特征对比

Table 1　Multi-dimensional feature comparison of dense and MoE models

对比维度	Deep-Seek-LLM-7B	Deep-Seek-R1-671B
基础架构	密集型模型	MoE 模型
总参数量/$\times10^9$	7	671
激活参数量/$\times10^9$	7	37
注意力机制	多头注意力	多头潜在注意力
典型并行策略	DP＋TP	TP＋PP＋EP
仿真域主导通信原语	AR	AG/RS，A2A

仿真中为两类模型配置了差异化的并行策略。对于 DeepSeek-LLM-7B，由于模型体积较小，在主流的训练显卡中没有极端的显存墙限制，因此采用 TP 与 DP 的混合策略。对于 DeepSeek-R1-671B，由于其巨大的静态权重和特征路由机制，采用显存切片与 EP 的混合策略，本文采用零冗余优化（Zero Redundancy Optimizer，ZeRO）显存切片技术，将模型的静态参数、梯度和优化器状态分块，均摊至集群内的各计算节点中，在计算时通过高速通信按需聚合与释放，降低显存开销。对两种模型的单轮训练负载进行建模后，本文统计了各类集合通信操作的流量占比，在 DeepSeek-LLM-7B 的流量中，AR 通信占主导地位，其余通信类型占比较低；在 DeepSeek-R1-671B 的流量中，AR 流量降低至不足 1%，AG 与 RS 合计占比 92% 的通信总流量，A2A 通信的占比也显著提升至约 7%。数据符合稠密模型和稀疏模型的通信特征：前者依赖全局梯度同步，后者因专家路由机制，通信重心转向更细粒度的数据收集分发和交叉交换。

2　基于稳态检测的外推加速算法

大语言模型在 DMT 时遵循固定的执行序列和通信计划，因此网络状态变化具有周期性。以 PP 为例：在预热阶段，网络带宽需求逐步上升，微批次按序注入，各节点分阶段向下游节点传输激活值。一旦流水线被完全填满，所有节点将同时执行前向与反向传播，带宽随之稳定在某一固定水平，在网络上表现为持久的长流[19]。进入冷却阶段后，随着最后一批微批次完成计算及流水线的排空，带宽会迅速降至零。

这种宏观上的周期性与稳态特征，同样存在于微观的流量行为中。现代大语言模型的混合并行架构中，微观通信主要由 TP 与 DP 驱动，本文将这两类并行策略下的通信操作建模为 P2P 流量时发现，所得流量呈现出高度的确定性与局部性：①每次训练迭代中的通信模式一致，触发时序与空间拓扑具有可重复性；②大量事件（约 90% 以上）集中在局部计算环内，机间通信的大部分流量单跳完成，不经过骨干交换机。③大小流特征明显，AR 流量大小约为 2 MB，RS 与 AG 流量大小约 100～200 MB。

基于上述大模型训练时的流量特征和网络特征，本文提出了一种基于稳态检测的外推加速算法。包含 3 个阶段，如图 3 所示，首先对每类通信操作的前 L 步 P2P 流执行细粒度的包级仿真，捕获网络真实的瞬时状态；其次引入自适应稳态监测引擎，追踪端到端时延和网络吞吐量等特征，一旦特征方差收敛至预设阈值，算法主动截断底层离散事件流，并启动稳态外推引擎，基于当前仿真数据对后续周期的网络状态和通信耗时进行数学外推。

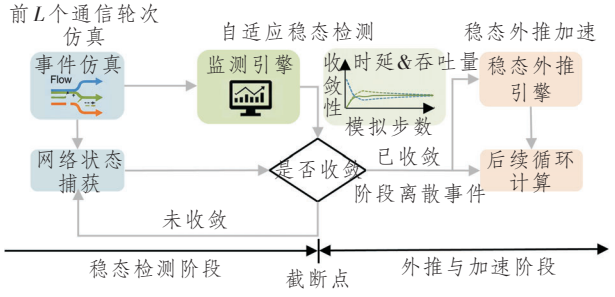


图 3　稳态检测外推加速算法概述

Figure 3　Overview of the steady-state detection and extrapolation acceleration algorithm

2.1　自适应稳态检测

基于 OMNeT++的事件驱动机制，在仿真控制层设置了滑动的观测时间窗 T_{win}，稳态外推引擎会

在 T_{win} 内对网络中所有活跃流 f_m 的瞬时吞吐量进行采样，记为 T_m，m 为活跃流索引。为了量化网络吞吐量的波动，计算连续窗口内吞吐量的极差 $v_{\text{th},m}$，定义其为样本集 T_m 中瞬时吞吐量的最大观测值与最小观测值之差，用于表征流在该观测时间窗内的速率振荡幅度，可表示为

$$v_{\text{th},m} = \max(T_m) - \min(T_m) < E_{\text{th}}, \tag{4}$$

式中，E_{th} 为吞吐量稳态阈值。当多个连续时间窗的瞬时吞吐量波动均$<E_{\text{th}}$时，判定网络在宏观流量分配上进入稳定状态。

在吞吐量趋于平稳后，网络中间节点仍可能出现突发拥塞或排队抖动。当系统在宏观流量分配上进入稳定状态后，监测引擎会对各流的端到端排队时延波动进行评估：

$$v_{\text{q}}^{k} = \max(Q_u^{k}) - \min(Q_u^{k}) < E_{\text{q}}, \tag{5}$$

式中：变动幅度 v_{q}^{k} 为第 k 个观测时间窗 T_{win}^{k} 内队列长度样本的最大值与最小值之差；Q_u^{k} 为在第 k 个观测时间窗 T_{win}^{k} 内，记交换机端口 u 的瞬时队列长度为量化队列的抖动程度；E_{q} 为队列稳定阈值。当 $v_{\text{q}}^{k} < E_{\text{q}}$ 时，判定网络在微观层面消除了剧烈震荡。

在OMNeT++仿真过程中，当上述双重稳态同时满足，且稳态持续超过2个观测周期，控制层才会正式确认最佳外推时机并进入外推阶段。

2.2　稳态外推

将开始触发外推的时间定义为 T_{start}，进入外推阶段后，仿真控制层会暂停底层节点包级事件的收发，基于当前的网络状态快照，将离散的报文交互抽象为连续的流体模型。具体而言，系统先获取 T_{start} 时刻活跃流 f_m 的待传数据量 L_m，随后，结合链路的可用容量与流的优先级权重，基于最大最小公平（Max-Min Fairness，MMF）原则计算出各流在稳态下的理论传输速率 V_{m}，离散的排队网络被映射为由线性方程驱动的速率模型。

外推阶段通过跳过冗余的仿真计算，实现仿真时钟的跨越推进。为了避免因跨度过大，仿真引擎越过数据流传输完成的时间节点，进而影响后续子流的发送，导致网络资源过早释放，打破现有的全局稳态，系统设置了安全外推步长 ΔT_{leap}，由最先完成传输的短板流决定。系统先计算活跃流在解析速率下的理论剩余完成时间 τ_m：

$$\tau_m = \frac{L_m}{V_m}。 \tag{6}$$

为保证仿真的连贯性并避免事件越界，系统选取所有 τ_m 中的最小值作为外推的有效时间步长，则 ΔT_{leap} 为

$$\Delta T_{\text{leap}} = \min f_m(\tau_m)。 \tag{7}$$

确定安全步长后，仿真引擎执行外推跃迁，将全局仿真时钟推进 ΔT_{leap}，跃迁至目标时刻 T_{end}：

$$T_{\text{end}} = T_{\text{start}} + \Delta T_{\text{leap}}。 \tag{8}$$

在该时间跨度内，系统基于 V_i 和 ΔT_{leap}，计算各流在 T_{end} 时刻的剩余数据量 L_{m_last}：

$$L_{m\_\text{last}} = L_m - V_m \cdot \Delta T_{\text{leap}}。 \tag{9}$$

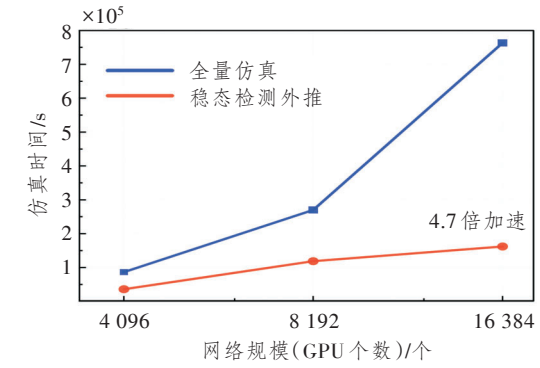
计算完成后，系统将 T_{end} 时刻的剩余数据量 L_{m_last} 写至应用层状态。对于满足 $L_{m\_\text{last}}=0$ 的流，判定其已完成传输并触发流完成事件，唤醒仿真控制层，启动新一轮的稳态评估与时间外推。

3　效果验证

本文基于OMNeT++离散事件仿真平台构建了3层胖树网络拓扑，对DeepSeek-R1-671B模型的P2P通信流量开展仿真实验。为全面评估加速算法的性能，实验设置了不同的网络规模（4k～16k），并从计算开销（模拟运行时间和内存占用）与保真度（网络时延和丢包率）两方面进行分析。随着集群规模的扩大，模型单次训练触发的通信事件呈指数增长，全流量仿真的计算开销超出可承受范围。针对此算力瓶颈，实验采用代表性切片验证方案：从全局通信流量中提取多段具备拓扑对称性与集合通信语义的流量作为测试用例，并固化为标准测试集，保证仿真输入为相同的流量模式和注入时序。实验中首先对该测试集进行两轮全量仿真，取其网络特性的均值作为对比基准；随后，启用本文所提稳态外推加速算法对同一测试集进行仿真，通过比对两者的计算开销与网络性能指标，实现对算法加速效果与保真度的量化评估。

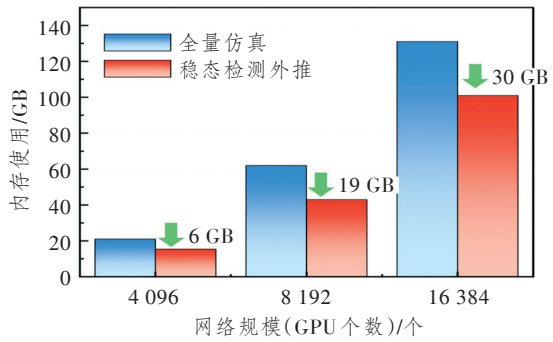
图4所示为分别在3种网络规模下使用传统全量仿真与稳态检测外推加速算法时，仿真过程中的计算开销对比。如图4(a)所示，针对DeepSeek模型在4k、8k和16k规模下的P2P通信流量，传统全量仿真时长呈类似指数的非线性增长，16k规模下仿真时长约为212 h，引入稳态检测外推加速算法后，增长趋势显著放缓，仿真时长降至约45 h，加速比为4.7倍。除时间开销外，内存占用同样是制约大规模仿真的主要因素，实验基于双路Intel Xeon Gold 6 148及192 GB内存的高性能服务器，利用Linux伪文件系统监测仿真任务全周期的内存占用量。如图4(b)所示，稳态检测外推算法在4k、8k和16k规模下，峰值内存占用分别降低6、19和30 GB，

有效降低了仿真的内存需求。实验结果证明，稳态检测外推加速算法降低了大规模网络仿真中的计算开销，提升了仿真效率。



(a) 仿真时长

(a) Simulation time



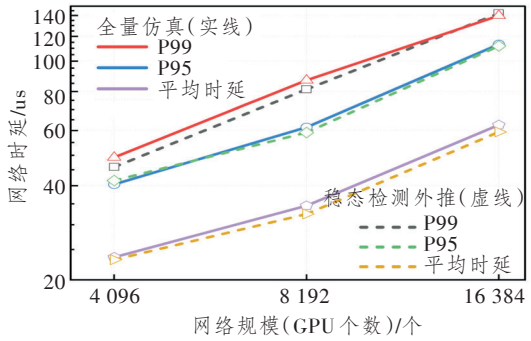
(b) 峰值内存占用

(b) Peak memory usage

图 4　仿真计算开销

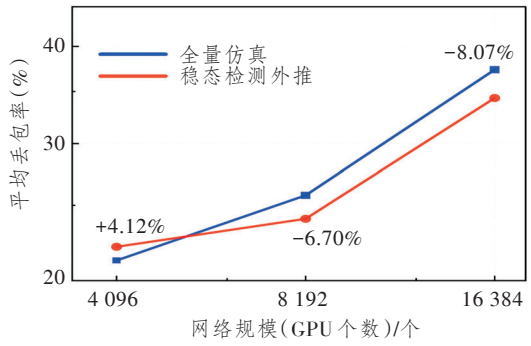
Figure 4　Computational overhead

图 5 所示为 3 种网络规模下，分别使用全量仿真和稳态检测外推加速算法，仿真结果的保真度对比。图 5(a)所示为 4k～16k 规模下两种仿真方法的时延指标，随着网络规模的扩大，平均时延和长尾时延均显著上升，但稳态检测外推加速算法的时延结果(虚线)与全量仿真的时延结果(实线)基本重合，两者偏差较小。从数据上来看，4k～16k 规模下，两种仿真方法平均时延的绝对偏差均小于 3.5 μs，相对偏差分别为 1.79%、5.88% 和 5.05%。由于 DMT 对同步高度敏感，最慢的少量数据包往往决定整体计算时间，因此，图中采用 P95 与 P99 时延刻画因局部拥塞导致的长尾效应，分别反映了网络中 95% 和 99% 的数据在传输时的最大时延界限。以采用全量仿真时的 P99 时延为例，4k～16k 规模下，时延由 45.86 增至 142.27 μs；除此之外，3 种规模下，P95 与 P99 的最大相对误差均控制在 7% 以内，在 16k 规模下，两者的绝对误差分别为 1.36 与 2.46 μs。图 5(b)所示为 4k～16k 规模下，两种仿真方法的平均丢包率。在 4k 规模下，稳态检测外推算法的丢包率高于全量仿真，这是由于小规模网络路径有限，易发生集中的瞬态拥塞。稳态检测外推算法在特征提取时，有限的统计窗口易捕获这类频发的拥塞事件；而在后续外推过程中，基于局部窗口计算出的高丢包率会被全局放大，导致外推值偏高。随着网络规模的增大，拓扑层级和可用路径增加，流量在空间分布上趋于分散，偶发的拥塞事件集中发生在稳态检测外推统计窗口内的概率显著降低，从而导致外推值偏低。两种方法的平均丢包率相对误差分别为 4.12%、6.70% 和 8.07%，整体偏差较低。实验结果证明，稳态检测外推加速算法在大规模网络仿真中具有较高的保真性。



(a) 网络时延

(a) Network latency



(b) 网络丢包率

(b) Network packet loss rate

图 5　仿真保真度对比

Figure 5　Comparison of average latency

真实智算网络通常依赖优先级流控机制(Priority Flow Control，PFC)或数据中心量化拥塞通知(Data Center Quantized Congestion Notification，DCQCN)等拥塞控制协议保障无损传输[20]，本文研究重点在于底层的稳态检测外推加速机制，为隔离上层协议干扰并检验稳态检测引擎的鲁棒性，测试采用无流控的原始网络配置，该配置下网络会产生严重的队列积压，导致高丢包现象。如图 5(b)所示，全量仿真在 4k 规模下的平均丢包率为 21.2%，

16k 规模下的丢包率升至 37.3%。为了对上述高丢包率现象进行归因，本文在 4k 规模下设计了补充实验。实验引入基于多级阈值的流控机制。该机制根据队列积压程度分两级响应：轻度积压时，触发显式拥塞通知（Explicit Congestion Notification，ECN）进行预警；重度积压时，交换机通过独立于业务流量的高优先级控制队列，向源端主动发送反压控制帧。实验结果表明，引入该流控机制后，4k 规模的网络拥塞问题得到控制，平均丢包率降低至 2.5%。

受物理内存制约，当前实验的最大验证规模为 16k，但本文所提稳态检测外推加速算法旨在实现十万卡规模的智算网络仿真。为此，本文通过流量抽样策略，对该算法在 130k 网络规模下的有效性进行了小规模仿真验证。具体而言，基于 DeepSeek-R1-671B 模型在 131 072 节点规模下的 A2A 通信操作进行流量建模，并提取 10 万条 P2P 流量作为代表性样本，在有限算力条件下完成了超大规模的仿真评估。结果如表 2 所示，在 130k 网络规模下，稳态外推算法相较全量仿真，仿真时间缩短了 73.0%，内存占用降低了 31.3%；同时，网络平均时延的相对误差为 11.9%，与全量仿真基准保持较高的一致性。验证了该算法在十万卡规模仿真中的有效性与保真性。

表 2　十三万卡规模网络仿真实验结果

Table 2　130 k-Scale Network Simulation Results

对比维度	全量仿真	稳态检测外推	相对差异
仿真时间/s	112 320	30 274	下降 73.0%
内存占用/GB	128	125	下降 31.3%
平均时延/μs	151	133	下降 11.9%

稳态检测外推加速算法在超大规模仿真下取得加速收益的本质，在于其对仿真事件数量的压缩。如图 6 所示，对比全量仿真需计算处理的 P2P 流量总数与引入加速机制后仿真器实际仿真计算的 P2P 流量数，3 种规模下的实际计算量分别降低了 58.49%、54.01% 和 61.43%，且随着网络规模扩大至 130k，集合通信触发的 P2P 流呈指数级爆发，系统能够以稳态外推方式压缩的流量数量也随之提升，在 130k 超大规模场景下，算法能够减少 50% 以上的冗余计算，具有更优的加速效果与扩展潜力。

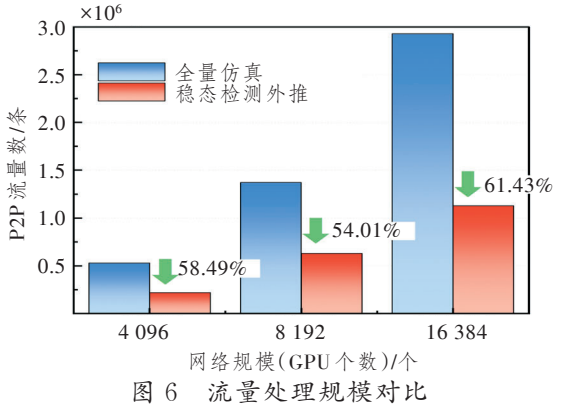


图 6　流量处理规模对比

Figure 6　Comparison of flow processing scale

4　结束语

本文针对超大规模智算集群网络性能评估中的算力挑战，详细分析了 DMT 的集合通信特征，构建了微观流解析框架。随后，以 DeepSeek 模型的微观流量为例，分析了底层 P2P 流量的同构依赖与时空规律。针对传统离散事件仿真的状态爆炸问题，基于流量周期性和网络的稳态收敛特性，提出了一种稳态检测外推加速算法。该算法在保持网络特征真实的前提下，实现了仿真效率的提升。本文分析的流量拆分机制与加速方法，为十万卡级别智算网络性能评估提供了参考。

参考文献：

[1] Feng S, Zhang J, Dai J, et al. Seamless Data Delivery for Distributed AI Workloads via Dynamic Queue Scheduling and FPGA-based Implementation in Converged Optical-Electrical Networks[J]. Journal of Optical Communications and Networking, 2025, 17(9): 820–833.

[2] Gui F, Gao K, Chen L, et al. Accelerating Design Space Exploration for LLM Training Systems with Multi-Experiment Parallel Simulation [C]//Proceedings of the 22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '25). Philadelphia, PA, USA: USENIX Association, 2025: 473–488.

[3] Li C, Zabreyko A A, Nasr-Esfahany A, et al. M4: a Learned Flow-Level Network Simulator [EB/OL]. (2025-03-03) [2026-05-14]. https://arxiv.org/abs/2503.01770.

[4] Wang W, Wu Z, Wang Y, et al. HyGra: Accelerating Network-State Simulation for LLM Training in DCNS via Adaptive Packet-Flow Granularity [EB/OL]. (2026-03-19) [2026-05-14]. https://arxiv.org/abs/2603.12671.

[5] Bai S, Zheng H, Tian C, et al. Unison: a Parallel-Efficient and User-Transparent Network Simulation Kernel[C]//Proceedings of the Nineteenth European Conference on Computer Systems. Athens, Greece: ACM,

2024：115–131.

[6] Gao K，Chen L，Li D，et al. DONS：Fast and Affordable Discrete Event Network Simulation with Automatic Parallelization [C]//Proceedings of the ACM SIGCOMM 2023 Conference. New York，NY，USA：ACM，2023：167–181.

[7] Long F，Gao K，Chen L，et al. Supercharging Packet-Level Network Simulation of Large Model Training via Memoization and Fast-Forwarding [EB/OL]. (2026-02-11)[2026-05-14]. https://arxiv. org/abs/2602. 10615.

[8] Narayanan D，Shoeybi M，Casper J，et al. Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM [C]//SC21：International Conference for High Performance Computing，Networking，Storage and Analysis. St. Louis，MO，USA：IEEE，2022：9910057.

[9] Rajbhandari S，Li C，Yao Z，et al. DeepSpeed-MoE：Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale [EB/OL]. (2022-08-21)[2026-05-14]. https://arxiv. org/abs/2201. 05596.

[10] Shoeybi M，Patwary M，Puri R，et al. Megatron-LM：Training Multi-Billion Parameter Language Models Using Model Parallelism [EB/OL]. (2020-03-13)[2026-05-14]. https://arxiv. org/abs/1909. 08053.

[11] Diéguez A P，Casellas Á B，Torres-Camps A，et al. Pretraining LLMS at Scale：Tuning Strategies and Performance Portability[C]//SC25-W：Workshops of the International Conference for High Performance Computing，Networking，Storage and Analysis. St. Louis，MO，USA：IEEE，2026：11358241.

[12] Liu A X，Feng B，Xue B，et al. DeepSeek-V3 Technical Report [EB/OL]. (2024-12-27)[2026-05-14]. https://doi. org/10. 48550/arXiv. 2412. 19437.

[13] 王冬柔，李新，赵辰宇，等. 面向大模型分布式训练的光组网技术综述[J]. 光通信研究，2025(6)：250050.
Wang D R，Li X，Zhao C Y，et al. A Review of Optical Grouping Techniques for Distributed Training of Large Models[J]. Study on Optical Communications，2025(6)：250050.

[14] Zhu H，Phanishayee A，Pekhimenko G. Daydream：Accurately Estimating the Efficacy of Optimizations for DNN Training [EB/OL]. (2020-06-05)[2026-05-14]. https://arxiv. org/abs/2006. 03318.

[15] Gibiansky A. Bringing HPC Techniques to DeepLearning[EB/OL]. (2017-02-21)[2026-05-14]. https://andrew. gibiansky. com/blog/machine-learn-ing/baidu-allreduce/.

[16] Yang H，Zhou H，Liu Z，et al. Energy Optimization of Wireless Sensor Embedded Cloud Computing Data Monitoring System in 6G Environment[J]. Sensors，2023，23(2)：1013.

[17] Bi X，Chen D L，Chen G T，et al. DeepSeek LLM：Scaling Open-Source Language Models with Longtermism [EB/OL]. (2024-01-05) [2026-05-14]. https://arxiv. org/abs/2401. 02954.

[18] Guo D，Yang D J，Zhang H W，et al. DeepSeek-R1：Incentivizing Reasoning Capability in LLMs via Reinforcement Learning [EB/OL]. (2026-01-04)[2026-05-14]. https://arxiv. org/abs/2501. 12948.

[19] 翟锐，李壮志，侯广营，等. 基于以太无损网络的智算中心光网络架构研究[J]. 光通信研究，2024(5)：240028.
Zhai R，Li Z Z，Hou G Y，et al. Research on Optical Network of Intelligent Computing Center based on Ethernet Lossless Networking[J]. Study on Optical Communications，2024(5)：240028.

[20] Li Y，Miao R，Liu H H，et al. HPCC：High Precision Congestion Control[C]//Proceedings of the ACM Special Interest Group on Data Communication. Beijing China：ACM，2019：44–58.