计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(9): 2401−2423, 2026

高性能智算中心网络:现状、挑战与趋势

张 娇[1] 刘怿睿[1] 康志远[1] 刘军卫[2] 孟令坤[2] 徐 军[2] 赵兴华[2] 季忠铭[2] 黄 韬[1]
[1](网络与交换技术国家重点实验室(北京邮电大学) 北京 100876)
[2](中移(苏州)软件技术有限公司 江苏苏州 215163)
(jiaozhang@bupt.edu.cn)

High Performance Artificial Intelligence Data Center Networking: Studies, Challenges and Trends

Zhang Jiao[1], Liu Yirui[1], Kang Zhiyuan[1], Liu Junwei[2], Meng Lingkun[2], Xu Jun[2], Zhao Xinghua[2], Ji Zhongming[2], and Huang Tao[1]
[1] (*State Key Laboratory of Networking and Switching Technology* (*Beijing University of Posts and Telecommunications*), *Beijing* 100876)
[2] (*China Mobile* (*Suzhou*) *Software Technology Co., Ltd.*, *Suzhou*, *Jiangsu* 215163)

Abstract Artificial intelligence data centers (AIDCs) have emerged as a new and increasingly critical form of computing infrastructure. Unlike traditional cloud data centers that primarily provide virtualization of general-purpose computing resources, AIDCs are designed to support high-performance, AI-centric workloads such as large-scale model training and inference. As AI models continue to grow and complex, AIDC networks are required to deliver unprecedented levels of bandwidth, low latency, and efficient cross-device coordination. We provide a comprehensive analysis of AIDC network characteristics, including service requirements, topology design, communication patterns, and traffic characteristics, and further examine the unique challenges that arise from these aspects. Building on a network layering perspective, we then present a structured overview of key enabling technologies for AIDC networks. These technologies include collective communication libraries, transmission control, load balancing, data-link flow control, and fault management. Representative academic studies and industrial implementations are summarized to illustrate their strengths, limitations, and suitability for real-world AIDC deployments. Finally, we outline several future development trends, such as the convergence of general-purpose and AI-oriented data centers, deeper specialization and co-design across AIDC hardware and software stacks, and the advancement of multi-tenant, shareable AI computing power. These trends are expected to play a pivotal role in shaping next-generation AI infrastructure.

Key words artificial intelligence data center networking; collective communication; transmission control; load balancing; fault management

摘 要 智算中心是数据中心发展到当前的一种新形态,与传统以提供虚拟化计算资源为主要特征的云数据中心相比,智算中心主要为以人工智能为代表的新型高算力需求业务提供强大算力。针对智算中心网络业务需求、拓扑结构、通信模式、流量特征等进行了深入阐述与分析,针对其独有特征带来的新问题与挑战逐一进行剖析。然后,按照网络分层模型梳理了以集合通信、传输控制、负载均衡、链路控制和故障管理为核心的高性能智算中心网络关键技术框架,并对这些关键技术现状和优劣势进行了详细的归纳

收稿日期: 2025-04-03; **修回日期**: 2026-05-12
基金项目: 国家自然科学基金项目(62132022); 山东省自然科学基金项目(ZR2023LZH011)
This work was supported by the National Natural Science Foundation of China (62132022) and the Shandong Provincial Natural Science Foundation (ZR2023LZH011).

总结。最后,在对当前技术发展分析的基础上,提出了通算智算一体化、智算中心技术专用化、智能算力多租户化等未来发展趋势。

关键词 智算中心网络;集合通信;传输控制;负载均衡;故障管理

中图法分类号 TP393

DOI: 10.7544/issn1000-1239.202550249 **CSTR:** 32373.14.issn1000-1239.202550249

当前,算力已成为推进我国数字经济高质量发展的重要基石[1-6],以大模型为代表的人工智能等业务对算力的需求急剧增长[7-8]。在这一背景下,算力基础设施以提供通用算力为主的传统云数据中心为基础[9-11],向提供专用算力为主的高性能智算中心持续演进[12-15]。智算中心(artificial intelligence data centers, AIDCs)是一种面向分布式计算的专用化数据中心,通过全新设计的异构算力架构,大范围采用 GPU 等专用芯片替代传统 CPU 通用计算芯片[16],实现诸如生成式大语言模型(large language models, LLMs)的训练、推理等工作[17]。智算中心能够针对特定计算任务需求进行优化,从而在相同的能耗下实现计算能力成倍增长,为以人工智能为代表的新一轮科技革命和产业升级注入强劲动力[12-15]。

算力需求激增和业务类型扩展要求网络中更多处在同一网络拓扑下的异构计算设备进行互联互通并实现高效的并行协同工作,而传统数据中心网络并不具备这一能力[12-15]。目前,传统数据中心内的核心计算业务所采用的典型叶脊(leaf-spine)拓扑正在向智算中心网络所采用的多轨道流量聚合拓扑架构演进[18],同时也涌现出如英伟达提出的 NVLink[19], NVSwitch[19] 等全新总线通信协议。

此外,传统的点对点通信原语已经很难满足人工智能业务对数据进行并行化处理的需求[20],为此业界设计了一套多对多集合通信原语并诞生了数据并行、流水线并行、张量并行等并行训练方式[21-24]。并且人工智能业务具有高度相似、可被量化建模的流量结构,使得智算中心流量的行为在一定程度上可以被预测,进而使智算中心网络具有可进行规律化描述和定量分析的业务流量特征,并基于此进行针对性的网络调度优化,专用于智算中心网络的集合通信库开发成为业界关注的一大焦点。

然而,智算中心网络的高速发展带来了很多新的问题与挑战。面对全新的业务应用模式,传统数据中心网络无法在异构并行的计算场景下实现跨节点联合计算。为了满足智算中心的异构并行计算需求,智算中心网络需要一套高效易用、可编程的集合通信库进行数据切分调度,实现计算进程与通信进程的高效流水并行;需要一种简便易行且高效可控的传输控制机制,尽可能避免网络中发生拥塞;需要采用可实现网络负荷稳定分摊并避免数据包乱序的负载均衡机制[25],从而充分利用冗余带宽资源实现网络负载的相对平均;还需要通过精细化的链路控制机制,在不影响网络吞吐和时延的前提下,满足智算中心网络在远程直接数据存取(remote direct memory access, RDMA)无损网络层面的需求。此外,在智算中心网络出现故障时,智算中心要能够及时发现并检查故障的根因,帮助运维人员尽快恢复至正常状态以免产生损失。

为了解决以上问题和挑战,智算中心网络领域涌现出一批全新的技术方案。以英伟达提出的英伟达集合通信库(NVIDIA collective communication library, NCCL)[26] 及基于 NCCL 的其他扩展通信库,如合成集合通信库(synthesized collective communication library, SCCL)[27]、拓扑感知集合通信库(topology aware collective communication library, TACCL)[28]、流量工程集合通信库(traffic engineering collective communication library, TE-CCL)[29] 等为代表的集合通信方式,为智算中心大规模参数同步任务提供了高可行性的解决方案;谷歌提出的 TIMELY[30], Swift[31], Poseidon[32] 等拥塞控制方案在避免智算中心网络拥塞的同时,实现了速率快速收敛、高稳定性和高公平性保障;以静态路由表为代表、配合故障诊断或软件定义网络(software defined network, SDN)控制器的方案则实现了智算网络内的流量均衡,最大化利用了网络内的冗余带宽;InfiniBand 协议[33] 中提出的信令(credit)机制利用“先授权后发送”的思想,实现了无损链路的精细化速率调控;以检测主机间故障的 Pingmesh[34] 方案和主机内故障的 Hostping[35] 方案为代表的故障管理方案,则为智算中心网络故障管理提供了丰富的解决措施。

未来,智算中心网络为了适应人工智能等高算力需求业务的高速发展将不断迭代演进,智算中心内需要传输的数据量日益激增,除了提升处理器算

力能力外，还需要更大的网络传输带宽、更高效的网络拓扑结构与通信协议，因此也对一体化分层专用方案设计提出了更高的要求。与此同时，作为一个人工智能时代的基础算力设施，智算中心应该具备被多方共享的能力，因此智能算力多租户场景将会是智算中心另一重要的新应用场景。最后，在数据中心向通算、智算、超算一体化智算中心演进的道路上，需要实现智算与传统数据中心组成的通算、超算一体化融合的过程，才能让智算中心具有更好的可部署性与健壮性。

1 智算中心网络特点

智算中心网络是一种服务于人工智能业务负载的基于异构硬件的网络架构，通过利用智算中心中典型分布式机器学习训练业务产生的流量特征、通信模式等特点对网络软硬件架构做适应性设计，以加速训练业务。

本节将对智算中心网络业务需求、拓扑结构、并行通信模式以及业务流量特征等核心特点分别进行介绍。

1.1 业务需求

智算中心以支撑大模型等人工智能业务为主。如图 1 所示，2020 年，OpenAI 公司推出了第 3 代预训练模型 GPT（generative pre-trained Transformer），即 GPT-3，模型参数规模达到了 1 750 亿[36]，成为当时最大的语言模型，并且在零样本学习任务上实现了巨大性能提升。随后，更多策略如人类反馈的强化学习（reinforcement learning from human feedback, RLHF）、代码预训练、指令微调等开始出现，被用于进一步提高推理能力和任务泛化。2023 年 3 月，OpenAI 发布的超大规模多模态预训练大模型 GPT-4 具备了多模态理解与多类型内容生成能力[37]。在迅猛发展期，大数据、大算力和大算法完美结合，大幅提升了大模型的预训练和生成能力以及多模态多场景应用能力[36-48]。

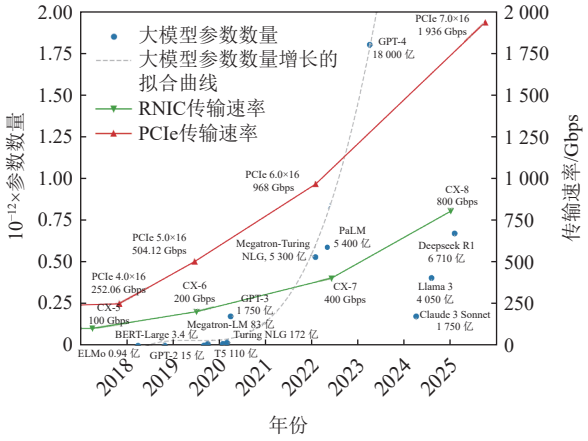


Fig. 1 Comparison of parameter scale and transmission rate for LLMs from 2018 to 2025[36-48]

图 1 2018—2025 年大模型参数规模与传输速率对比[36-48]

从业务需求上来看，大模型的参数规模由最初的 9 400 万爆发增长至 1.8 万亿，参数量按照每 2 年约 400 倍的速度增长，需要万卡级别的 GPU 集群才能容纳下 GPT-4 这样的大模型；同时，参数量的迅速增长造成通信开销的等比例增长，而总线互联带宽增速却仅为每 2 年约 1.4 倍。此外，GPU 等待网络通信的空闲时间为 25%~68%，导致大量的 GPU 计算资源被浪费。在 GPT-4 的训练过程中，能够保证 GPU 不被闲置的期望单向网络带宽为 800 Gbps 或 1.6 Tbps，而目前最先进的 RDMA 网卡 NVIDIA ConnectX-8 的单向网络带宽速率为 800 Gbps，大规模应用的主流网卡带宽仅为 200 Gbps 或 400 Gbps，远低于 GPT-4 的 GPU 无空闲期望带宽值。因此迫切需要对硬件拓扑以及通信策略 2 个方向进行优化，从而使网络能够提供更大的带宽服务以及优化数据通信和减少通信开销，从而更好地满足大模型训练日益激增的带宽需求。

1.2 拓扑结构

数据中心网络主要由三大网络架构组成：计算结构拓扑、存储结构拓扑、管理网络拓扑。随着 RDMA 技术的成熟应用和与日俱增的数据中心算力需求，数据中心网络需要协同跨节点间的计算设备进行联合计算来提供更高算力，传统数据中心网络在计算、存储、管理网络中同时采用的叶脊拓扑等树形结构已然不能满足当前计算业务的需求，智算中心网络着重对计算结构进行改进，提出了多轨道流量聚合架构。以阿里巴巴公司设计的 HPN 智算中心网络为例[49]，HPN 由前端网络和后端网络组成，前端网络负责管理和存储等其他流量，后端网络主要承载训练过程中计算产生的流量，采用 2 层双平面架构，相比传统 3 层 Clos 架构，减少了网络跳数和等价多路径路由（equal-cost multi-path routing, ECMP）哈希次数，提高了路径选择的精确性。主机内部采用的是多轨道流量聚合架构，为每台服务器配置与 GPU 数量相同的网卡，以满足业务训练时巨大的跨服务器通信流量需求。

如图 2 所示，多轨道流量聚合架构将不同节点上位于相同位置的 RDMA 网卡（RDMA network interface card, RNIC）连接在同一叶交换机上。自上而下的每

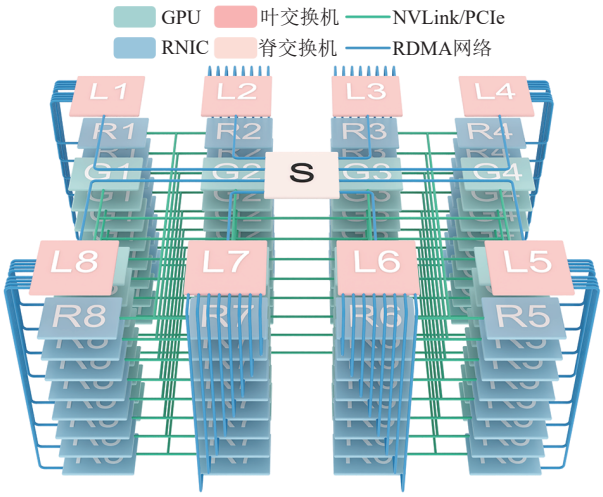


Fig. 2 The multi-track network architecture of AIDCs

图 2 智算中心多轨道网络架构

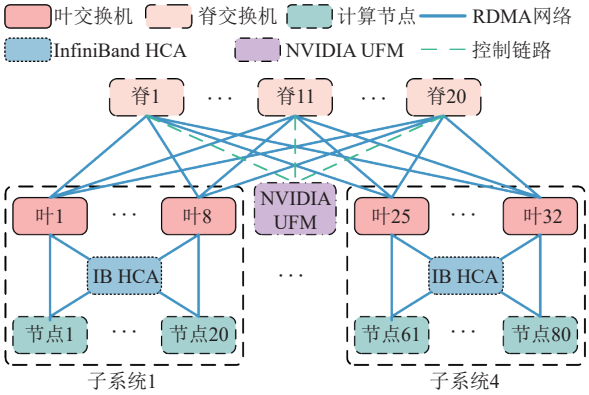


Fig. 3 Illustration of computation architecture topology for 80 nodes in DGX SuperPOD

图 3 DGX SuperPOD 80 个节点计算结构拓扑示意图

一层均包含独立的节点 1~8, 每个节点内部含有 8 个 GPU 与 8 张网卡, 所以整个计算网络在跨节点互联上划分为 8 个独立并行的轨道平面, 每个平面上采用 RDMA 进行互联(例如节点 1~8 的 RNIC 1 通过叶 1 组网), 而节点内通过 PCIe 或 NVLink 进行互联(例如节点 1 的 GPU 1~8)。假设节点 1 的 GPU 1 需要与节点 2 的 GPU 2 进行通信, GPU 1 可以先将数据通过 NVLink/PCIe 发送至节点内的 RNIC 2, 然后 RNIC 2 通过轨道平面 2 发送至节点 2 的 RNIC 2, 最后在节点内将数据发送至 GPU 2, 从而充分利用了节点内的高带宽链路, 并且使跨节点传输只需要经过 1 层交换机。在多轨道网络架构中, 大模型训练产生的通信需求, 如全局归约(all-reduce)、全对全(all-to-all)、全局聚合(all-gather)等操作都可以利用多个轨道并行传输加速, 并且大部分流量都聚合在轨道内传输(只经过叶交换机), 小部分流量才会跨轨道传输(需要经过脊交换机), 大幅减轻了大规模并行训练下的网络通信压力。

如图 3 所示, 标准的 DGX SuperPOD[50] 计算架构可以采用 80 个 GPU 节点完成组网, 在每一台脊交换机上预留的特定端口以连接英伟达统一架构管理器(unified fabric manager, UFM), 通过增强的实时网络遥测和 AI 驱动的网络智能分析功能实现对网络流量的整体感知与集中优化, 并确保 DGX A100 系统上的每个 InfiniBand 主机通道适配器(host channel adapter, HCA)都通过胖树(fat-tree)[51] 拓扑结构将 GPU 节点接入网络, 从而有效避免了网络拥塞问题。

1.3 并行通信模式

过去几年来分布式深度学习模型的规模急剧增加, 更多的数据量与更大的模型参数量对于单个 GPU 来说资源量太大, 无法塞入整个训练模型, 所以业界越来越多的采用成千上万张 GPU 卡进行并行训练。模型训练过程中伴随波动性、周期性的模型参数、梯度等相关数据的通信, 其需要超过半数甚至全部服务器参与通信。为了更好地利用分布式计算资源, 优化数据传输模式, 减少通信量和通信次数, 提高系统的带宽利用率, 产生了诸多并行通信模式, 如数据并行(data parallelism)、流水线并行(pipeline parallelism)、张量并行(tensor parallelism)、混合专家(mixture-of-experts, MoE)并行等。

数据并行[21] 遵循单进程多数据(single program multiple data)的原则, 即每个进程维护相同的模型参数和相同的计算任务, 但是处理不同的数据集(data batch)。通过这种方式, 同一全局数据(global batch)下的数据和计算被切分到了不同的进程中。

在传统数据并行下, 每个进程也会持有对参数的完整拷贝, 占用了大量内存。ZeRO 通过将模型的参数、梯度和优化器状态划分到不同 GPU 进程, 在训练前后按需进行参数的收集和释放, 从而消除冗余的内存占用。

如图 4 所示, 流水线并行[22] 对模型中间层进行横向拆分并分配给 4 个 GPU 进行训练, 在此基础上流水线并行进一步将小批次(mini batch)数据切分成更细粒度(4 份)的微批次(micro batch)数据, 在微批次数据中采用流水线式的执行顺序, 在训练过程中, 数据按照顺序在不同阶段的 GPU 之间流动。前一个阶段的 GPU 计算完成后, 将输出作为下一个阶段 GPU 的输入。例如, 第 1 个阶段的 GPU 对输入数据进行计算, 然后将结果传递给第 2 个阶段的 GPU 继续处理, 依此类推, 直至最后一个阶段完成计算。在

每个训练迭代周期中，各阶段的GPU有明确的同步屏障，保证同一GPU的正向计算和反向计算都拥有相同的模型参数，确保数据的有序处理，从而可以将GPU计算资源利用率从25%提升至57.1%。对于拥有m个设备的流水线并行处理过程，随着小批次数据拆分数n越来越大，GPU的利用率将会越来越高，达到$n/(n+m-1)$，但是，过大的n值会带来参数陈旧性的问题，影响训练收敛速度。

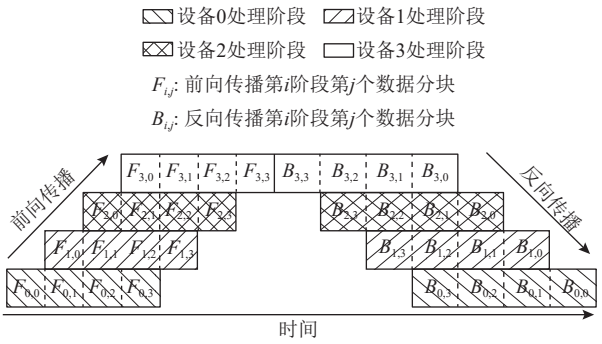


Fig. 4 Illustration of acceleration principle of pipeline parallelism

图4 流水线并行加速原理示意图

如图5所示，张量并行[23]对模型中间层纵向拆分，即每个Transformer层内的矩阵乘法被分割到多个GPU上并行计算，每块GPU计算完成后，进行矩阵归约相加，其结果等价于原矩阵乘法。

如图6所示，混合专家并行[24]由门控网络和专家网络2个部分构成。门控网络负责根据输入数据的特征，动态地决定哪些专家模型应该被激活以生成最佳预测。专家网络是一组独立的模型，各模型有分别擅长的子任务。通过门控网络，输入数据将被分配给最适合的专家模型进行处理，并根据不同模型的输出进行加权融合，得到最终的预测结果。

DeepSeekMoE[52]是一种混合专家架构，旨在最大限度地提高专家模型的专业化程度，尤其是在大语言模型中。它主要通过2种策略来实现这一目标：

1）细粒度专家分割（fine-grained expert segmentation）。传统的混合专家架构（如GShard）通常激活N个专家中的前K个，这导致知识混合性和冗余性问题。因为每个专家分配到的token可能涵盖不同的知识，专家被迫学习不同类型的知识，难以有效利用。同时，不同专家学习的知识可能重叠，造成参数冗余。DeepSeekMoE将专家数量从N扩展到mN个更小的专家，同时激活mK个专家。这样做保持了参数总量和计算成本不变，但增加了激活专家的组合灵活性。这种细粒度分割允许将不同知识更精细地分解并分

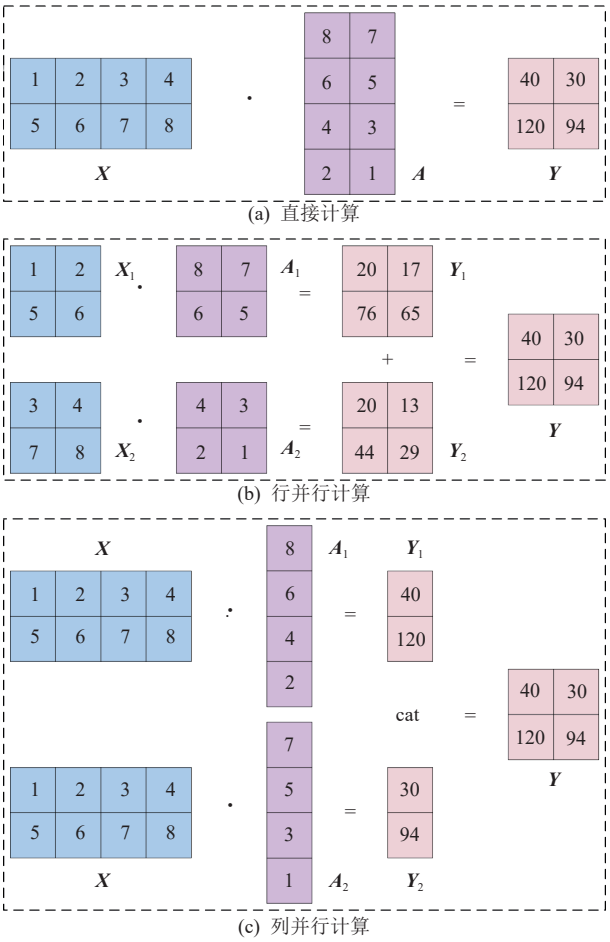


Fig. 5 Illustration of tensor parallelism principle

图5 张量并行原理示意图

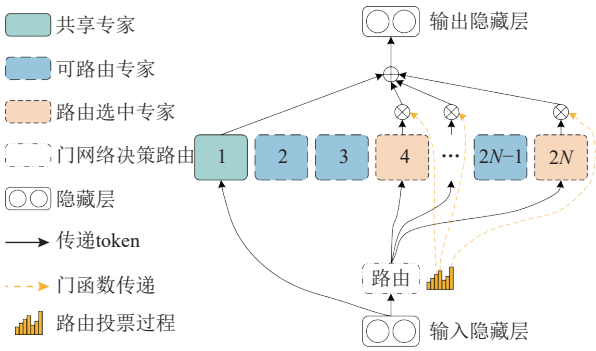


Fig. 6 Illustration of MoE parallelism principle

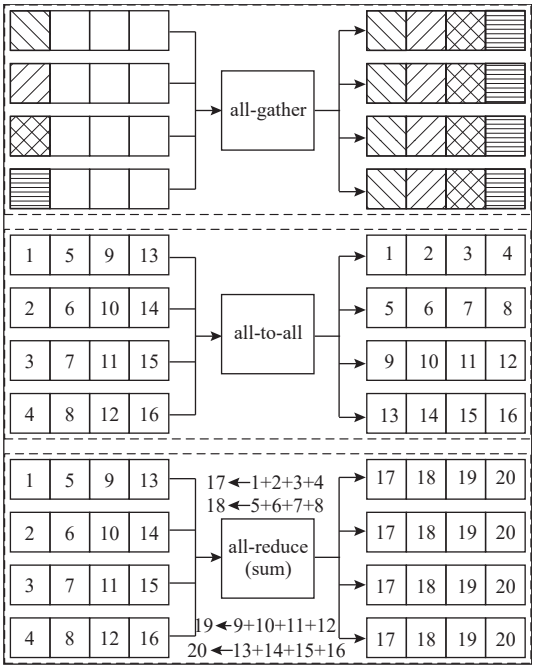
图6 混合专家并行原理示意

配给不同的专家，使每个专家更专注于特定领域，从而提高专业化程度。

2）共享专家隔离（shared expert isolation）。DeepSeekMoE隔离出K个共享专家，这些专家始终处于激活状态，负责学习和整合不同上下文中的通用知识。通过将通用知识压缩到共享专家中，可以减少路由专家之间的冗余，使路由专家更专注于学习特定领域的知识，从而提高专业化程度和参数效率。

1.4 业务流量特征

当前智算中心网络中的业务存在大量单对多、多对多等流量模式,其最为基础的操作有发送、接收、广播、分发、节点内进程栅栏同步以及节点间进程同步,这些操作经过组合构成了基本通信原语[28]。如图 7 所示,常用的集合通信原语有:all-gather, all-to-all, all-reduce 等。



注:图中的数字表示不同的纹路,非实际数值。

Fig. 7 Examples of data in four GPU buffers before and after typical collective communication primitive operations

图 7 典型集合通信原语操作前后 4 块 GPU 缓冲区数据示例

原语 all-gather 的操作是聚合(gather)+广播(broadcast)的组合,收集多个进程中相同类型的数据并广播至所有进程中。原语 all-to-all 是对数据的多对多转置,其过程涉及到单个节点对所有节点的分散(scatter)以及所有节点对单个节点的聚合(gather)。原语 all-reduce 操作是归约(reduce)+广播(broadcast)的组合,将多个进程中的数据按照约定好的规则进行运算以生成单个数据并广播至所有进程当中。

数据并行中的通信过程发生在梯度全局更新 all-reduce 操作中。每块计算 GPU 将自己的梯度传输给梯度收集 GPU 做聚合操作。这里的聚合操作一般指梯度累加求和或求平均。相较于数据并行需要 GPU 之间的全局通信,流水线并行只需在 GPU 间点对点通信传递部分层间参数,这样的特性可以使流水并行对通信带宽的需求降到更低。张量通信发生在每层的前向传播和后向传播过程中,类型为 all-reduce,不但单次通信数据量大,且通信频繁。所以张量并行一般都在机内训练,通过高速 NVLink[19] 来进行加速,流水线并行一般都在机间训练,通过 InfiniBand 交换机进行连接[53]。如图 8 所示,混合专家并行的通信过程主要发生在门控网络以及各专家将参数全局更新至所有专家的 all-to-all 操作过程中[54]。

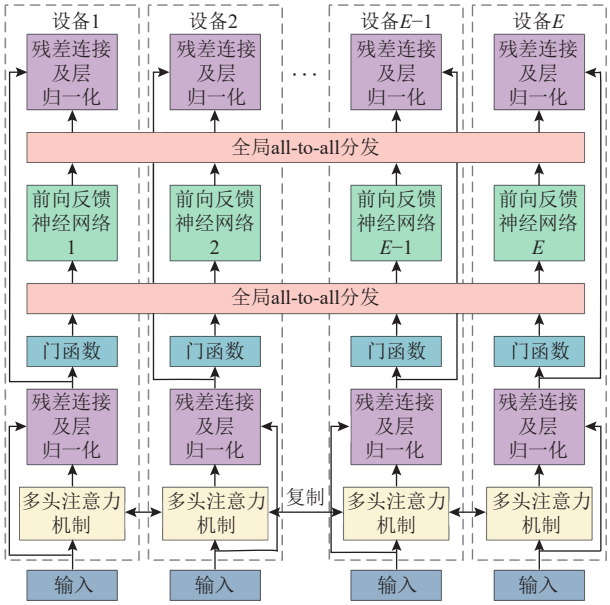


Fig. 8 The process of all-to-all in MoE parallelism

图 8 混合专家并行的 all-to-all 过程

大模型等业务的训练模型产生了通断(on/off)业务流量。如图 9 所示,on/off 模型[55] 只有通(on)和断(off)两个状态,当业务处于 on 状态时以恒定速率传输数据,off 状态不产生任何数据。通信流仅产生在 GPU 计算训练后得到的参数传递到各 GPU 的过程中,而跨 GPU、服务器训练会带来大量的网络通信开销[56-59],GPU 等待网络通信的时间会占整体训练时间的 20%~65%[39,60],造成了计算资源的极大浪费。实现最佳的训练效率的理想状态是内部计算进程与节点间通信进程均处于繁忙状态,以流水线的形式工作,形成图 9 中浅色虚线所示的模式。该模式可通过挖掘训练过程中数据的并行性,将数据划分成多个可执行的块并均匀分布到计算进程以及通信进程中来实现[61-62]。

2 智算中心网络面临的挑战

在新兴业务需求的驱动下,智算中心网络为了承载超高吞吐的业务流量,采用了全新的拓扑结构

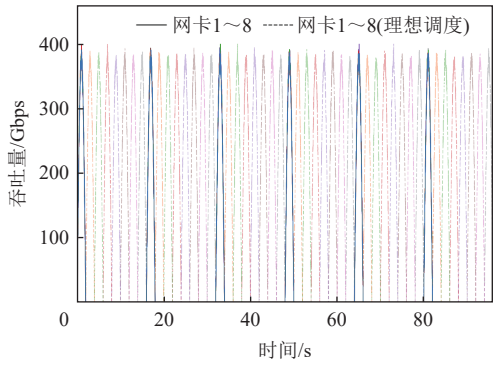


Fig. 9 on/off traffic generated from LLMs training

图 9 大模型训练产生的通断流量

和多并行通信模式。同时，很多人工智能模型训练业务会产生大规模、多周期的峰值 on/off 流量，这既是机遇也是挑战，其周期性可以降低网络通信过程的熵值，使得其具有更强的规律性以便进行预测和调度，而其大规模并发性则会对网络产生极大的压力。如果不对智算中心所承载的此类业务进行针对性优化，将会造成智算中心网络算力资源的极大浪费，这对于硬件成本造价高昂的智算中心而言是不可接受的。相比传统的数据中心网络，智算中心网络面临着诸多全新的挑战，本节将会对智算中心网络面临的这些挑战与难题逐一展开介绍。

2.1 多 GPU 间通信效率低

在本文 1.1 节中介绍了人工智能大模型的规模在过去几年间急剧增加导致单个 GPU 的存储资源和计算资源都难以满足大模型的训练需求，因此业界普遍采用跨 GPU/服务器集群进行大模型训练。然而，大规模的分布式 GPU 训练服务器集群并不能保证算力与规模之间呈线性增长关系。影响性能提升的主要挑战有：

挑战 1.1. 在节点内 GPU 互联中，PCIe, NVLink[19] 表现出较强的非一致性内存访问（non-uniform memory access, NUMA）架构效应[63]，单个 CPU/GPU 内的访存性能显著高于跨 CPU/GPU 访存性能，存在主机内访存性能瓶颈。

挑战 1.2. 异构设备的互联互通存在适配难度。例如，NVLink[19] 在带宽上比 PCIe 具有更大优势，而在延迟上并没有同样的优势；节点内通信的带宽和时延性能均显著优于节点间通信，但单一节点内处理的数据量有限。对开发者友好的开发模式应通过多层级的应用程序编程接口（application programming interface, API）调用，将面向业务的代码与面向异构设备适配的代码分离，从而能够分别实现业务逻辑与底层性能适配的目标。

挑战 1.3. 所有深度学习任务均呈现计算-通信交替的 on/off 周期性。迭代内先进入“计算阶段”（GPU 满负荷，无网络流量，流量 off 态），再进入“通信阶段”（GPU 等待参数/梯度同步，产生网络流量，流量 on 态），且同一任务的周期在迭代间高度重复。当多个任务处于通信阶段 on 态时，会竞争同一链路带宽，导致通信延迟显著增加，即 GPU 需等待更长时间才能完成同步，进而延长“空闲窗口”。

挑战 1.1~挑战 1.3 均对实现高效 GPU 间通信提出了较高要求。为此，需要高效的集合通信技术来承接对设备的统筹管理和异构适配任务，详细的内容将在 3.1 节中展开介绍。

2.2 交换节点网络拥塞

急剧增加的大模型规模需要更大规模的智算中心网络来支撑，而更大规模的智算中心带来了更复杂的互联结构和更高速的流量传输挑战，这就造成了拓扑中交换节点的网络拥塞现象。具体而言，解决这种拥塞现象所面临的主要挑战有：

挑战 2.1. 周期性训练数据传输时会产生大规模的并发流量，大量流量瞬间涌入交换节点会造成网络拥塞，极大地增加了流平均完成时间，从而降低了 GPU 算力资源的利用率。

挑战 2.2. 现有的传输控制算法大多需要在起始阶段流量增长速率的快速收敛和拥塞避免之间做出权衡，这将会造成在链路空闲时带宽资源的浪费。尽管现有的传输控制算法通常能够保证较快的增长速率，但对于智算中心网络的高价值算力资源而言，需要更加高效的传输控制策略。

挑战 2.1 和挑战 2.2 表明，为传统传输控制协议（transmission control protocol, TCP）网络的数据中心设计的传输控制算法并不能完美适用于智算中心。为此，需要使用为 RDMA 网络和智算中心设计的传输控制算法来适应智算中心的低熵流量特点，详细的内容将在 3.2 节中展开介绍。

2.3 带宽资源利用低效

为了保障网络的可靠性、带宽的冗余性，智算中心网络任意 2 个节点之间通常有多条转发路径。然而，智算中心业务在实际使用这些转发路径时则很难实现充分利用，其面临的主要挑战有：

挑战 3.1. 以整流粒度进行负载均衡的方案对于长短流混合的传输过程，会造成链路利用率不均衡，即不同链路上传输了相同数量的流量，但由于流量长度不同而造成了部分链路过载其他链路闲置，造成带宽资源利用率下降。

挑战 3.2. 以数据包粒度进行负载均衡的方案虽然能够很好地实现负载均衡的目标,但对于大量采用 RDMA 技术,特别是 RoCE 技术的智算中心来说,会因数据包到达路径不同导致数据包乱序,从而导致对数据包乱序极为敏感的 RoCE 协议产生大量的丢包重传,降低有效吞吐量。

挑战 3.1 和挑战 3.2 在传统的数据中心中是无法兼顾的,许多方案根据其应用场景选择了适合的平衡点。进入智算时代,智算中心网络需要兼顾链路利用率和数据包的按序到达率,使得这二者能够同时得到解决。对这类方案的介绍将在 3.3 节中展开介绍。

2.4 无损网络传输性能未达预期

为了实现智算中心网络无损传输的目标,除了高可靠的物理层介质降低了偶然丢包和误码的可能,还需要可靠的数据链路层控制机制来确保网络传输的无损性。然而,现有的无损网络传输机制的设计会对网络性能产生较大影响,其面临的主要挑战有:

挑战 4.1. 无损网络需要数据链路层的流控机制来处理传输控制失效时造成的网络拥塞事件,但是其作为“兜底”方案的反应速度和调节效果都弱于传输控制。

挑战 4.2. 通过为交换设备配置暂停上游数据发送的门限阈值虽然能够实现无损网络的流控机制,但是会在较长时间内无法发送数据,甚至会造成局部乃至全局的死锁并终止数据传输,造成吞吐量迅速下降甚至归零,严重影响网络性能。

挑战 4.1 和挑战 4.2 为智算中心网络实现更有效的无损网络控制方式提出了较高要求。解决方案将在 3.4 节中展开介绍。

2.5 故障检测与定位困难

自 2018 年初以来,大模型的参数规模已经增长了近 2 万倍,训练模型所使用的智算中心也由原先的数百块单集群 GPU 增长到了上万块多集群 GPU。日益增长的参数规模需要不断扩增的智算中心加以存储、计算,进而造成如今的智算中心规模十分庞大,为此也对智算中心网络故障检测与定位提出了如下挑战:

挑战 5.1. 大规模的智算中心需要分钟级甚至秒级的快速定位能力,以减少由于设备故障而造成的算力资源浪费,降低智算中心运行和维护成本。而制约定位和探测速率的主要原因是网络通信开销和处理算力开销。

挑战 5.2. 繁杂的网络拓扑和异构算力设备对故障原因的判断产生了极大的挑战。故障管理方案必须具有详尽完善的故障知识储备才能够快速识别判断故障原因,从而启用对应的解决方案。

挑战 5.3. 上万块多集群 GPU 智算中心建设成本极为高昂,难以通过部署较多热备份设备保障服务的可靠性,同时也不便频繁中断业务进行全面检修。因此这需要完备的解决方案保障业务服务质量,例如通过平滑过渡到性能降级运营模式最大程度减少故障所带来的损失。

因此,一套完备的故障管理系统应该具备解决挑战 5.1 至挑战 5.3 的能力,并帮助运维人员对智算中心运行状态进行实时监测、及时报警、发现故障根因、保障业务服务质量。相关解决方案将在 3.5 节中展开介绍。

3 智算中心网络关键技术

为了解决第 2 节中智算中心网络面临的诸多挑战与问题,在智算中心网络协议栈的不同层中,都采用了专为智算中心设计的全新关键技术,为智算中心网络运行提供了重要保障。图 10 展示了在网络协议栈不同层中智算中心网络涉及的关键技术,本节将对集合通信、传输控制、负载均衡、链路控制及故障管理这 5 项关键技术逐一展开介绍。

3.1 集合通信

集合通信是一种有组织的多对多通信形式,在并行计算、分布式计算和深度学习应用场景中无处不在,是智算中心网络应用层中的关键技术。集合通信抽象了许多复杂的通信流程,能够实现数据的聚合并向多个进程传播,同时保留相对简单的 API。而集合通信库是用于实现集合通信操作的一种软件工具或库,它提供了高层次的抽象接口,对集合通信操作进行封装和实现,隐藏了底层通信的细节,使得开发人员能够更加方便地使用集合通信操作来实现数据交换和同步,有效简化分布式训练过程。集合通信库有效实现了对面向异构设备的适配,从而能够让开发人员专注于业务代码的开发,实现了业务与设备实现的分离。

在学术界,消息传递接口(message passing interface, MPI)[64] 是涉及集合通信的大规模科学应用中使用最广泛的编程模型之一,然而并未在工业界被广泛采用。许多行业领先的企业,如英伟达和微软等,都为深度学习应用开发了自己的集合通信库(collection communication library 或 collective communication library, CCL),将经典的 MPI 实现演变为它们

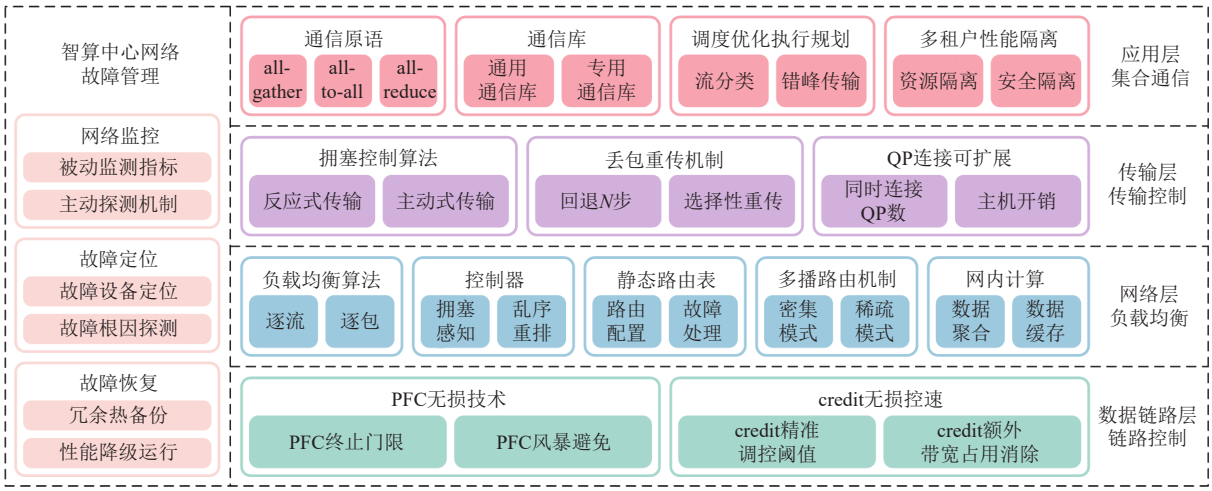


Fig. 10 The key technology architecture of AIDCs network

图 10 智算中心网络关键技术架构

的机器学习/深度学习(machine learning/deep learning, ML/DL)版本。

英伟达公司设计了基于 NVIDIA-GPU 设备的 NCCL[26], 专用于机器学习或深度学习的工作负载。其编程模型与 MPI Collective 非常相似。与 MPI[64] 专注于在集群中数千个节点之间的高效通信不同的是, NCCL 致力于在密集的多 GPU 系统中提供 GPU 之间高效、可扩展的快速消息通信, 采用统一计算设备架构(compute unified device architecture, CUDA)执行 GPU 之间的通信[65], 使用环形算法在所有 GPU 之间传输数据[66], 并在通信过程中按块分割数据以保证高带宽利用率和低延迟。NCCL 通过 PCIe[67], NVLink[68], GPUDirect 实现节点内通信, 通过 GDR(GPUDirect-RDMA)[53] 实现节点间通信, 实现了异构连接方式的统一控制。同时, NCCL 采用了与 MPI 类似的 API, 实现了对原有项目最大程度地兼容。NCCL 有效解决了挑战 1.1 和挑战 1.2, 实现了对异构设备和通信方式的优化, 使得开发人员无需关注底层实现即可实现业务需求, 这些特性也让 NCCL 成为深度学习应用中集合通信的首选库。

然而, NCCL 最突出的问题是其作为英伟达配合其 GPU 产品推出的集合通信库, 具有很强的专用性和排他性, 对于挑战 1.3 的解决程度有限。为此, 涌现出了一批与 NCCL 相似但更加开放包容的集合通信库算法以解决挑战 1.3。

阿里巴巴集合通信库(Alibaba collection communication library, ACCL)是阿里巴巴自研通信库[69], 其基于成熟的 PCIe 和网卡 NIC 技术, 提出了一种物理网络拓扑 BiGraph, 在每个服务器内配备了与 GPU 等量的网卡(network interface controller, NIC), 并将每一 GPU 和 NIC 进行绑定配对, 从而解决了 NVLink[19], NVSwitch[19] 可扩展性差以及不兼容其他 GPU 的缺点, 并针对该拓扑提出了 all-reduce 操作的一种混合算法实现[70](节点内 PCIe 采用 ring 算法, 节点间以太网通信采用 HD 算法), 设计将多个算法共存于一个单一的集合通信操作中。

腾讯集合通信库(Tencent collective communication library, TCCL)是一款针对腾讯云星脉网络架构的通信库[71], 其基于星脉网络硬件架构搭建, 具备网络故障快速感知与自愈的智能运维能力。TCCL 基于 NCCL[26] 在网卡设备管理、全局网络路由、拓扑感知亲和性调度、网络故障自动告警等方面进行扩展优化, 从而更好地适用于星脉网络。

微软集合通信库(Microsoft collective communication library, MSCCL)[72] 提出了适用于多节点拓扑的合成集合通信算法, 其解决了 NCCL[26] 只适用于节点内拓扑的集合通信算法生成的缺点。如图 11 所示, MSCCL 将交换机与网卡的互联抽象成逻辑直连链路 Switch-hyperedges[28]。基于探测机制生成逻辑直连链路的带宽与延迟, 并支持用户灵活选择链路连接策略, 将算法合成器扩展至节点间互联。

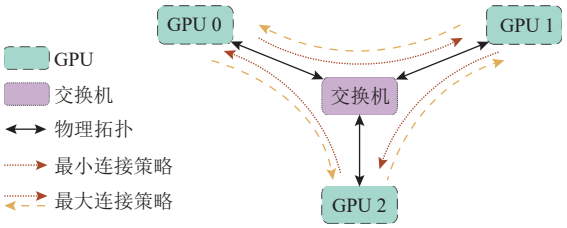


Fig. 11 Switch-hyperedges logical links

图 11 Switch-hyperedges 逻辑链路

为了适应多节点异构的智算中心网络计算场景,集合通信库不断迭代,通过对数据块实现更为细粒度的切分控制和加强集合通信操作的拆分,构建出了操作灵活、适应性更广的集合通信算法,提高计算单元的利用率,实现计算进程与通信进程的高效流水并行,向着可编程、可扩展、更细粒度控制的方向发展。另一类针对大规模集群设计的跨任务通信调度优化技术例如 Crux[73], CASSINI[74] 等以“最大化 GPU 利用率”为核心目标,通过感知深度学习任务的流量特性(周期性、GPU 强度),在多任务间协调通信路径、时间、优先级,从源头避免通信竞争,实现了“网络调度”与“深度学习任务特性”深度融合。

Crux[73] 围绕“多任务通信竞争导致 GPU 利用率低下”的核心问题展开。它通过引入“GPU 强度”(计算量/通信时间)指标量化任务对集群效率的影响,优先为高 GPU 强度任务分配最不拥挤的网络路径,以避免链路竞争,结合迭代时间、计算-通信重叠特性微调任务通信优先级,并通过基于 DAG 切割的优先级压缩算法适配硬件有限优先级级别,同时动态响应任务启停并进行重新调度。该技术从源头减少多任务间的通信竞争,解决了传统调度忽略任务价值差异、路径选择随机、硬件限制制约调度效果等问题,最终实现 GPU 利用率提升 8.3%~23%,显著降低高价值任务的通信延迟与 GPU 空闲时间。

CASSINI[74] 针对“多任务通信阶段重叠引发链路拥堵、GPU 等待时间过长”的关键问题,利用深度学习任务 on-off 流量的周期性特征,将每个任务的通信模式映射为几何圆(圆周长=迭代时间),通过旋转圆(对应任务通信起始时间偏移)实现不同任务通信相位的交错,再通过二分亲和图与广度优先搜索(breadth-first search, BFS)遍历计算全局唯一时间偏移,确保多链路任务在所有链路的通信相位均能协同,同时基于兼容性分数优化任务放置。该技术解决了传统调度缺乏全局通信相位协同、动态任务场景下调度时效性失效等问题,有效减少链路拥堵与显式拥塞通知(explicit congestion notification, ECN)标记(最多降低 97%),使任务平均迭代时间与尾部迭代时间分别缩短 38% 和 60%,提升集群整体训练效率。

这一类方案针对多节点异构的智算中心网络计算场景进行单独设计,在满足挑战 1.1 和挑战 1.2 的同时,利用了智算中心网络特性努力解决了挑战 1.3 带来的问题。

3.2 传输控制

传输控制是在智算中心网络协议栈传输层控制数据中心流量发送方式的关键技术,主要由 2 种功能不同的控制技术构成:一是拥塞控制,旨在通过调节发送端的发送速率,保证网络内瞬时数据量不会过载,确保网络的可访问性和通畅性[75-77];二是丢包重传,旨在对网络中由于各种因素未能成功传递至接收端的数据进行重新传输,确保传输信息的可靠性和完整性。本节将对这 2 个方面逐一展开介绍。

3.2.1 拥塞控制

智算中心网络拥塞控制的核心目标是解决高并发流量下的网络拥塞问题和实现高网络吞吐量以及低传输延迟。表 1 对比了智算中心网络常见的拥塞控制方案。在采用 RDMA 技术的智算中心网络中,目前被最广泛采用并部署的拥塞控制方案是第 2 代基于以太网的 RDMA 技术 RoCEv2(RDMA over converged Ethernet version 2)协议设计的数据中心量化拥塞通知(datacenter quantized congestion notification, DCQCN)[78],其核心思想是由交换机根据端口队列排队长度依概率标记 ECN,接收端在收到带有 ECN 的数据包后通过生成拥塞通知数据包(congestion notification packet, CNP)通知发送端进行降速,发送端则根据是否接收到 CNP,确定发送速率应当如何调整。DCQCN 最突出的特点是简单实用易部署,并在一定程度上解决了挑战 2.1,但其最大的问题是速率稳定性差,发送速率收敛速度慢,对于挑战 2.2 的解决程度有限。

为了更加精细地调控智算中心网络拥塞情况,业界还诞生了针对 RoCEv2 协议和类 RoCEv2 协议设计的基于往返时延(round-trip time, RTT)的智算中心网络拥塞控制方案。该类方案的代表是谷歌公司提出的 TIMELY[30], Swift[31], Poseidon[32] 系列方案和阿里公司提出的 HPCC[79] 等。Mittal 等人[30] 提出的 TIMELY 能够在网卡硬件中准确测量时间戳的精细化 RTT 测量模块,通过计算梯度预测网络拥塞的变化情况,采用数据包发送间隔代替了拥塞窗口(congestion window, cwnd)加性增长乘性减少(additive increase multiplicative decrease, AIMD),实现了更加精细的发送速率控制方案,对挑战 2.1 的解决程度较 DCQCN 有一定改善,但对挑战 2.2 中提出发送速率收敛慢的问题并未得到有效解决。

如图 12 所示,Kumar 等人[31] 提出的 Swift 在 TIMELY[30] 的基础上进一步明确了端侧拥塞和链路拥塞,详细定义了链路拥塞的各组成部分,引入了目标时延缩放机制以获得更加准确的 RTT 测量结果,此外还重新启用了 AIMD 的 cwnd 大小调整方式,增

Table 1 Comparison of Common Congestion Control Algorithms for AIDCs Network

表 1 智算中心网络常见拥塞控制算法对比

算法	控制方式	适用协议	基本思路	初始速率控制	常规速率控制	其他特点
DCQCN[78]	响应调整	RoCEv2	ECN 降速	逐渐上升	AIMD	依概率标记 ECN
TIMELY[30]	响应调整	RoCEv2	根据 RTT 调速	逐渐上升	梯度控制	精准 RTT 测量低速率精准控制
HPCC[79]	响应调整	RoCEv2	根据 RTT 调速	线性速率	达到限制后停止	收集 INT 信息
Swift[31]	响应调整	Pony Express	根据 RTT 调速	逐渐上升	AIMD	高精准 RTT 测量低速率精准控制
Poseidon[32]	响应调整	Pony Express	根据 RTT 调速	自适应上升	自适应 MIMD	同时增速或减速低速率精准控制 最拥塞 INT 收集
ExpressPass[80]	主动传输	InfiniBand	基于 credit 控制	0	credit	反馈式速率调节
Homa[81]	主动传输	InfiniBand	基于 credit 控制	线性速率	credit	计划外数据包优先
Aeolus[82]	主动传输	InfiniBand	基于 credit 控制	线性速率	credit	计划内数据包优先
MLTCP[83]	调度调整	TCP	感知发送数据	逐渐上升	调节拥塞窗口	
MLT[84]	响应调整	VMA	丢弃低优先级数据包缓解拥塞	线性速率	梯度控制	端到端系统控制允许有损传输
PLOT[85]	响应调整	TCP	丢弃低优先级数据包缓解拥塞	线性速率	线性速率	协议层面改进丢失容限控制重传 允许有损传输
CASSINI[74]	调度调整	RoCEv2	亲和图计算最佳延迟发送时间	根据调度延迟发送	线性速率	预先评估发送模式引入几何评估 方式
Dynamic Scheduling[86]	调度调整	TCP	获取流量调度信息和链路负载	高负载链路延迟发送	根据负载动态调整	考虑流量依赖关系

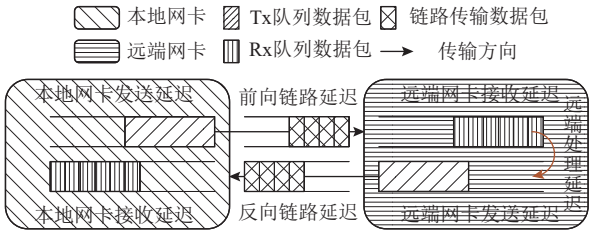


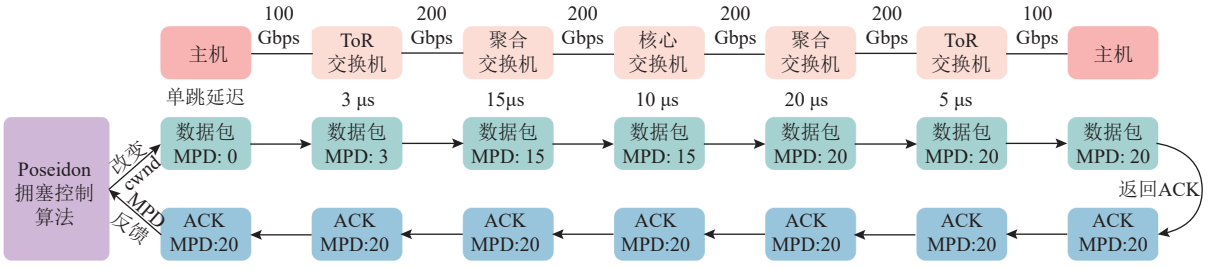
Fig. 12 Allocation method of end-to-end RTT in Swift scheme

图 12 Swift 方案端到端 RTT 划分方式

加了极小发送速率时的发送时间间隔设定，以满足精细化调整发送速率的需要。Swift 对于挑战 2.1 和挑战 2.2 都有一定程度的贡献，但仍具有改进空间。

Li 等人[79]提出的高精度拥塞控制（high precision congestion control, HPCC）方案利用了带内网络遥测（in-band network telemetry, INT）技术，准确收集了交换机队列长度以体现当前网络内的数据量，发送端可以减少发送数据量以保持网络内正在传输的数据总量始终保持在时延带宽积的附近。HPCC 与 Swift 方案具有类似的性能，但采用了不同的技术以实现目标。

如图 13 所示，Wang 等人[32]提出的 Poseidon 在 Swift[31]的基础上，使用 INT 技术获取网络内最拥塞的瓶颈节点交换机队列长度，并对瓶颈节点的最大速率流降速，实现不同速率流在同一瓶颈节点相同队列长度下的非同步速率控制。Poseidon 使用了自适应步长下乘性增长乘性减少（multiplicative increase multiplicative decrease, MIMD）代替 AIMD 对 cwnd 大小进行调整，并对每条流采用稳定性控制机制，确保了多流传输的公平性，降低了智算中心网络流平均完成时间。Poseidon 比 Swift 更好地解决了挑战 2.2 中提出的流量增长速度和拥塞避免的权衡问题，同时还通过对不同流量的自适应调控改善了 Swift 对挑战 2.1 的解决程度。



注：MPD 为最大单跳延迟。

Fig. 13 The information collection method of INT for Poseidon scheme

图 13 Poseidon 方案 INT 信息收集方式

此外，还有利用 InfiniBand 协议[33]的 credit 机制的拥塞控制方案。该类方案的代表是 ExpressPass[80]，Homa[81]，Aeolus[82]。如图 14 所示，Cho 等人[80]提出的 ExpressPass 方案采用 credit 控制发送端速率，在交换节点根据数据大小和 credit 大小比例计算得到的预留带宽限制 credit 的发送速率，使发送端收到的 credit 数量代表可用带宽总量。ExpressPass 虽然有效控制了网络拥塞，比较好地解决了挑战 2.1 的问题，但对于 credit 到来前的网络带宽利用不充分，相较于 DCQCN 对挑战 2.2 的解决能力更弱。

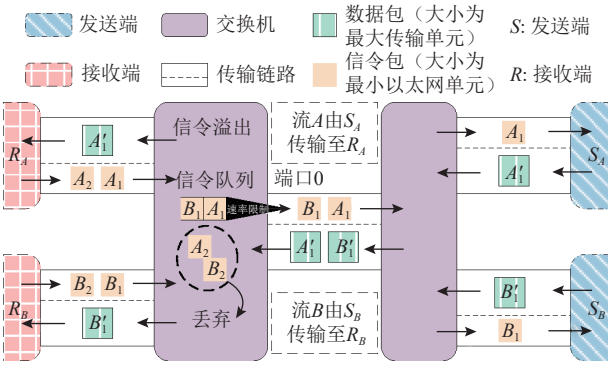


Fig. 14 Bandwidth allocation logics in ExpressPass scheme

图 14 ExpressPass 方案带宽分配逻辑

Montazeri 等人[81]提出的 Homa 方案利用 credit 到来前的空闲时间积极传输数据，以充分利用带宽，但 Homa 过于激进的传输策略导致 credit 控制的数据包发送效果有所欠缺，在智算中心低熵网络流量下解决挑战 2.2 的同时，忽视了挑战 2.1 中的高并发流量拥塞问题。

Hu 等人[82]提出的 Aeolus 方案结合了 2 个类别方案的特点，积极利用 credit 到来前的空余带宽的同时，优先保障基于 credit 的数据传输，实现了对空余带宽利用和网络拥塞控制之间的有效平衡，对挑战 2.1 和挑战 2.2 均有不错的解决效果。

上述方案都是由 TCP 数据中心切换至 RDMA 智算中心过程做出的重要尝试，通过对流量发送速率的精细化调控和流量增长速率的高效准确控制，实现了较为出色的拥塞控制效果。部分方案还通过对不同流的适应性调控以及对拥塞信号的精确获取，取得了更加出色的拥塞控制效果。同时，借助于 InfiniBand 协议 credit 机制的拥塞控制方案也可以让拥塞控制过程变得更加精细准确。

但是，上述方案并没有完全利用智算中心内训练流量的低熵特性，近年来在学术界的前沿研究领域中，也涌现出了部分针对智算中心的专用拥塞控制方案。以 MLTCP[83]，MLT[84]，PLOT[85]，CASSINI[74]，Dynamic Scheduling[86]为代表的方案有效地利用了智算中心的训练流量特征，结合训练流量的特点和数据包的重要性，形成了全新的拥塞控制思路。

Rajasekaran 等人[83]提出的 MLTCP 方案从应用层获取了大模型训练的每次迭代过程中要发送的字节数，并根据此来调整通信流的 TCP 拥塞窗口大小，从而让大规模并发的通信流在带宽争用时实现时间间隔交错，减少了网络拥塞情况。MLTCP 通过对发送数据的感知，有效地解决了挑战 2.2，并缓解了挑战 2.1 带来的影响，但并不兼容支持内核旁路的 RDMA 协议。

Wang 等人[84]提出的 MLT 方案是一套面向端到端整体网络传输过程的智算中心网络优化方案，其中包括的拥塞控制算法借助大模型训练数据传输的特点对于数据包的重要性进行了分类，在网络拥塞时优先传输较为重要的早期层梯度和较大的梯度数据，重要性较低的晚期层梯度和较小的梯度则会排队等待或者被丢弃，从而缓解拥塞问题。MLT 基于 VMA 协议开发，该协议可以拦截 TCP 的底层套接字调用，将其转换为对应的 RDMA 接口调用，减少传统应用适配 RDMA 智算中心的成本。Lu 等人[85]提出的 PLOT 方案集中于协议方面的改进，使用了改进的 TCP/UDP 混合协议传输机制，为不同层的梯度数据设计了细粒度的丢包容限，通过丢失非必要数据缓解网络拥塞情况，并对重要的控制数据包进行了传输保护。该类方案能够比较有效地同时解决挑战 2.1 和挑战 2.2 带来的问题，但是并不是传统意义上的无损拥塞控制方案，适用范围具有严格的限制。

Rajasekaran 等人[74]提出的 CASSINI 方案使用亲和图对通信流量进行几何抽象，除了在集合通信层面优化了智算中心网络的流量调度以外，在拥塞控制方面也能够通过预先评估作业兼容性及最佳延时发送时间，交错执行训练作业计算和通信过程，减少网络拥塞从而提升训练速度。CASSINI 相比于先前提出的 MLTCP[83]方案增强了对流量的调度和调控能力，并增加了对 RoCEv2 协议的支持。Zhao 等人[86]提出的 Dynamic Scheduling 方案通过和交换机通信获取流量调度信息和链路负载估计值，在考虑机器学习流量依赖关系的情况下最小化并发流量之间的竞争时间，动态调整流量发送速率，从而最大化网络带宽利用率并加速训练进度。这一方案虽然基于 TCP 完成，但是增加了对 NCCL[26]的兼容性支持。该类方案是无损拥塞控制方案，能够比较有效地同时解决

挑战 2.1 和挑战 2.2 带来的问题，但同样会受到适用条件的制约。

3.2.2 丢包重传

目前主流的丢包重传机制主要包括回退N步（go back N, GBN）和选择性重传（selective retransmission, SR）。GBN 的设计更为简单，不需要为乱序数据包提供缓冲空间，一旦数据包发生乱序或者丢包，将会浪费额外带宽进行重传。SR 的设计则需要缓冲区来存储乱序数据包，以确保在收到不连续数据包时等待缺失序号数据包传输。

这 2 种丢包重传方式虽然自 TCP 协议出现后就陆续被提出，但由于在采用 RDMA 技术的无损智算中心网络中出现偶然丢包的概率较低，因此在早期的智算中心网络建设过程中为了简化网卡设计主要采用的是使用 GBN 技术的网卡。然而，随着智算中心网络速率的高速发展，等待重传的时间内发送的数据量日益提升，采取 GBN 的丢包重传机制会浪费大量的带宽，在一个典型 RTT 为 2 μs、链路速率为 400 Gbps 的智算中心中，每次触发 GBN 重传都将会浪费 100 KB 的传输流量，并且可能会由于重传而加剧网络拥塞，造成更严重的影响。因此，学术界和产业界都开始探究采用 SR 技术的 RDMA 网卡。

在学术界，Mittal 等人[87]提出的增强的 RoCE 网卡（improved RoCE NIC, IRN）方案在 RoCE 网卡中实现了 SR 机制，通过在接收到乱序数据包时发送含有缺失数据包的累计序列号和触发重传的乱序数据包序号的否定确认（negative acknowledgement, NACK）以通知发送端进行数据重传，接收端会缓存所有乱序数据包进行等待，发送端维护了 1 个位图以跟踪数据包的接收情况。这种机制有效地减少了丢包重传时的数据量，减少了智算中心网络重传带来的负荷。

Huang 等人[88]提出的 FaSR 丢包重传机制针对 RDMA 网络通过快慢路径分离（单包丢失走无需位图的快速路径，多包丢失启用位图的慢速路径）、二级共享状态池（共享 SR 状态池与位图池动态分配资源），实现了在 RNIC 芯片上的完全卸载，并达到 200 Gbps 以上的处理速率，既大幅降低单连接内存开销（平均每连接仅 10 b），又保障高并发（5 000 以上连接数）下的低时延与高吞吐量，解决传统 GBN 低效、IRN 内存占用高、SRNIC 依赖 CPU 时延过大的问题。

在产业界，英伟达公司也发布了 ConnectX 系列的 ConnectX-6 Dx, ConnectX-7, ConnectX-8 等型号的 RNIC 和 BlueField 系列的 BlueField-2, BlueField-3 等型号的数据处理器（data processing unit, DPU）和超级网卡（SuperNIC），在采用 RDMA 协议的网卡上启用了 SR 技术从而减少重传数据量。

在学术界，与 MLT[84], PLOT[85] 等拥塞控制方案配套提出的丢包重传机制根据流量传输特点，不再要求严格无损传输网络，使得网络拥塞时可以以较低的代价选择性丢弃部分优先级较低的数据包，并在特定的时刻对其中部分丢包数据进行重传。同时，这些方案对于梯度数据的无序性进行了考量，进一步降低了端侧处理乱序数据包的丢包重传任务的难度。

丢包重传方式的改进不仅让智算中心拥塞控制算法拥有更加灵活的设计方式，还能够解决逐包负载均衡和链路控制算法中的很多局限性。SR 在 RDMA 网卡中的实现有效地帮助一些拥塞控制、负载均衡方案解决 2.2 节和 2.3 节中涉及的相关挑战。

3.3 负载均衡

负载均衡是在智算中心网络协议栈中网络层确定数据中心流量转发方式的关键技术，旨在将流量分摊到多条路径进行发送，以提升网络吞吐量，增强网络的灵活性和可用性[89]。表 2 展示了常见的智算中心网络负载均衡算法。

Table 2 Comparison of Common Load Balance Schemes for AIDCs Network

表 2 智算中心网络常见负载均衡方案对比

方案	类型	划分依据	基本思路	乱序情况	稳定性
ECMP[90]	动态均衡	逐流	五元组哈希	无	高
MP-RDMA[91]	动态均衡	拆分子流	将单一流量拆分为多条子流配合多路径 ACK 时钟和同步机制	子流间不保证顺序	高
ConWeave[92]	动态均衡	类逐包簇	依据信号在 2 条路径中切换	接收端 ToR 重排	高
MLT[84]	静态均衡	逐包	端侧添加标签指定转发路径无包依赖的梯度数据无需保持顺序	无需处理数据包顺序 应用层将数据包归位	高
Spectrum-X[93]	感知调控	逐包	端网协同动态监测	较少	低
RALB	感知调控	逐包/逐流	识别分类处理预定流量静态路由，突发流量 ECMP	较少	高
静态路由+故障诊断	静态路由	逐包/逐流	出现故障时切换回 ECMP	几乎没有	高
静态路由+SDN 控制	静态路由	逐包/逐流	实时监测可用路径更新路由表	几乎没有	高

在采用 RDMA 技术的智算中心网络中，目前被最广泛采用并部署的负载均衡算法的是为 RoCEv2 协议设计的 ECMP[90]，其核心思想是由交换机根据源 IP 地址、源端口号、目的 IP 地址、目的端口号、网络协议的五元组通过某种算法计算得到哈希值，确定转发端口。ECMP 是一种逐流负载均衡算法，保证了数据包到达接收端不会乱序，但对于不同的链路带宽和流长没有做出区分，因此在大小流混合的场景下实现的均衡效果不够稳定，对于挑战 3.1 的解决程度有限。

在学术界，负载均衡的改进主要集中在开发更细粒度的交换节点负载均衡算法，同时避免数据包乱序。Chen 等人[91]提出的 RDMA 多径传输（multi-path transport for RDMA, MP-RDMA）方案为 RoCEv2 协议的数据中心网络设计了用于多路径传输的三大机制：采用多路径 ACK 时钟机制，以逐包 ACK 反馈为依据动态分配流量，无需维护每条路径的独立状态，实现拥塞感知调度；通过乱序感知路径选择算法，主动剪枝慢路径、筛选时延相近的快速路径，配合 64 b 小型位图跟踪数据包，将乱序程度控制在可接受范围；提供同步机制保障关键场景下的内存更新顺序，同时兼容多数场景的直接数据写入以避免性能损耗。MP-RDMA 的本质是在发送端采用了多路径的传输方式，将单一数据流量主动拆分为多个路径传输，之后再交由接收端将多径流量重新合并。方案虽然能够较好地同时满足挑战 3.1 和挑战 3.2 中提出的问题，但依赖的随流 ACK 时钟机制和同步语义接口对网络条件有很严苛的要求，部署成本较高。

Alizadeh 等人[94]提出的拥塞感知均衡（congestion aware balancing, CONGA）采用包簇的思想，将流内连续数据包视为包簇，根据包簇表判断转发路径。在提升同一路径上不同链路的平均利用程度的同时，由于不同包簇之间间隔时间较长，因此数据包从不同转发路径转发时潜在的乱序风险相对可控，对于挑战 3.1 和挑战 3.2 采取了一种折中的解决方案。

Ghorbani 等人[95]提出的分布式随机网络内本地化负载均衡（distributed randomized in-network localized load-balancing, DRILL）将数据包逐包分配至各路径中队列最小的队列中，进一步提升了对链路的平均利用程度，但也进一步加剧了数据包乱序的程度，在更彻底解决挑战 3.1 的同时，也使挑战 3.2 的问题更加突出。

Qureshi 等人[96]提出的保护性负载均衡（protective load balancing, PLB）通过类包簇的数据包划分方式，在流空闲时随机切换路径，相比于 CONGA 的包簇表固定路径转发的方式，PLB 具有更好的非拥塞链路感知与选择能力，使得在对挑战 3.2 的解决能力与 CONGA 相近的情况下，对挑战 3.1 具有更好的解决效果。

Song 等人[92]提出的拥塞编排（congestion weaving, ConWeave）则在源端 ToR 设置了 2 条互为备用链路的路径转发方式，在目的 ToR 通过不同的队列实现了乱序数据包的重排序，从而以类包簇的形式实现了接近任意粒度的拥塞感知负载均衡机制。这一方案既利用了逐包负载均衡的有效均衡效果，又保证了与逐流负载均衡时相似的按序传输能力，实现挑战 3.1 和挑战 3.2 的同时解决。

Wang 等人[84]提出的 MLT 方案除了包含拥塞控制算法以外，还提出了针对智算中心网络负载均衡的算法改进措施。MLT 指出智算中心网络模型训练传输时的流量是由大量 32 b 单精度浮点梯度数据构成的，梯度数据之间不存在依赖关系，无需保持数据包顺序，因此可以容忍乱序数据包。MLT 会在端侧显式指定数据包转发路径，并添加梯度数据对应的位置标识信息用于在应用层恢复数据包的顺序，从而实现更优的负载均衡效果。该方案能够同时解决挑战 3.1 和挑战 3.2 带来的问题，但具有严格的使用场景限制，并且会增大传输时用于控制信息的带宽开销。

在产业界，对于负载均衡的探索则更多集中在如何利用现有负载均衡算法特点和智算中心网络流量特征在发送端进行针对化配置。英伟达的 Spectrum-X[93]解决方案通过网络侧交换机和终端 DPU 之间的紧密耦合和协调，实时动态监测每个 ECMP 链路的物理带宽以及端口出口的拥塞情况，在接收端接收数据前进行乱序数据包重排，从而实现了按数据包进行动态负载均衡，即逐包的自适应路由机制。Spectrum-X 与 ConWeave 的核心思路都确保了接收端前有乱序数据包重排机制，但 Spectrum-X 更好地利用了英伟达整套解决方案的优势，还具有链路状态的动态感知能力。Spectrum-X 同样能够同时较好地解决挑战 3.1 和挑战 3.2，但与 ConWeave 相比，其局限在于依赖英伟达的全套设备解决方案。

锐捷的 RALB 方案则根据感知到的网络拥塞情况，具有数据包类型的识别能力，将智算中心大模型训练时的流量进一步地分割，实时改变其负载均衡路径，从而实现了逐包的负载均衡机制；而对于其他类型的数据包，则采用传统的 ECMP 负载均衡方式，

以保证这些数据包的正常传输。RALB 方案根据对数据包的划分，能够分别采用逐流和逐包的负载均衡方案以同时应对挑战 3.1 和挑战 3.2。

此外，还有一些公司选择了预先配置的静态路由方案，直接配置不同数据包的转发路径。这种方案在流量具有高度规律性的智算中心中可以获得出色的效果，但当智算中心网络中的转发设备发生故障时，静态路由无法按照既定路径转发，此时就需要能够快速切换到备用方案。Meta 公司在静态路由的基础上增加了实时故障诊断机制，当检测到原路径发生故障时，自动切换回 ECMP，以保证智算中心网络的可用性；英伟达和华为则通过 SDN 控制器维护每台交换机的静态路由表，确保在原有静态路由路径发生故障后实时将路由表更新为可用转发路径。这些方案的共同特点都是对流量的种类进行了区分，充分利用了智算网络低熵流量的特性，为这些流量采用预设的负载均衡方式从而同时保障数据包到达顺序和吞吐量；同时，还设置了性能降级措施，以确保故障发生时网络的正常运行。

上述方案通过不断优化的负载均衡机制，充分融合了逐包与逐流负载均衡的优势，使得网络层在进行负载均衡时能够有效应对智算网络的高吞吐流量发送。部分方案还通过在接收端接收数据前进行数据包重排序来消除逐包负载均衡产生的乱序数据包，从而在最优的吞吐量下实现无乱序的数据包负载均衡。此外，产业界的成熟解决方案还考虑了对拥塞状态的动态感知与故障时性能降级的保障措施，进一步加强了智算中心网络高吞吐流量传输的可用性。

3.4 链路控制

链路控制是在智算中心网络协议栈中数据链路层控制数据中心流量发送许可方式的关键技术，旨在通过特定的控制信号，在交换机队列堆积时，降低上游交换机发送数据的速率，从而缓解队列堆积。链路控制的主要思路有 RoCEv2 协议使用的优先级流量控制（priority-based flow control, PFC）机制[97]和 InfiniBand 协议[33]使用的 credit 机制。

RoCEv2 的 PFC 机制的核心思想是“先发送后暂停”。在交换机特定端口缓冲区占用较高时，暂停其上游交换机低优先级数据的发送过程，待缓冲区有充足的空间后再恢复这部分数据的发送。根据设定的阈值，当缓冲区占用超出暂停阈值时，向上游交换机发送 PFC 暂停信号，上游交换机未发送的数据在端口缓冲区内排队等待，并逐级传递；当缓冲区低于恢复阈值时，向上游交换机发送 PFC 恢复信号，端口数据传递恢复正常。PFC 机制作为一种为与其配套使用的 DCQCN 拥塞控制算法的“兜底”机制，能够通过暂停发送的方式确保交换机不丢失数据包，但可能会因为 PFC 的传递性产生 PFC 风暴，降低数据传输效率。此外，还可能会由于多个节点的暂停发送造成 PFC 死锁，进而造成整个网络无法传输数据。通过合理配置 DCQCN 和 PFC 对应参数的阈值能够基本解决挑战 4.1，增强链路控制的灵敏程度，但是仍然会带来挑战 4.2 中的问题。

InfiniBand 的 credit 机制的核心思想则是“先授权后发送”。发送端在发送数据前，通常需要先发送 credit 请求数据包，获得发送的许可。接收端在收到 credit 请求后，会根据网络状况发送一定数量的 credit 包，以表明其可接收和缓存的数据总量。当所有可用的 credit 都已经被发送后，接收端将停止发送新的 credit，直至接收到新的数据将被占用的 credit 额度释放出来后，才会继续发送 credit。credit 机制的出现，一方面降低了发送端与接收端协商速率的难度，发送端只需按照接收端传递的 credit 数量，发送对应的数据即可完成发送；另一方面，credit 的动态调控可以让发送端准确获知网络内剩余带宽，有助于上层网络协议根据网络剩余带宽进行拥塞控制，降低了拥塞控制算法的设计复杂度。这种机制结合适当的参数配置，可以有效解决挑战 4.1，并且避免挑战 4.2 的问题。但是 credit 本身需要消耗一定带宽进行传输，因此将 credit 与现有的其他数据包（如 ACK）进行复用，成为降低 credit 开销的一种必要方式。此外，智算中心内现有的大量不支持 InfiniBand 协议的旧设备可以通过不断迭代、逐步替换的方式由 PFC 机制平稳切换至 credit 机制。

链路控制方案的改进除了能够有效保证数据中心无损网络的运行状态，还能够丰富传输控制方案的设计思路。credit 机制“先授权后发送”的思路为智算中心实现低时延、高吞吐的大规模数据传输提供了基本保障，能够帮助传输控制更好地解决 2.2 节中涉及的相关挑战。

3.5 故障管理

随着智算中心承载的业务规模不断增大，智算中心的分布式系统架构包含数万台设备共同执行预定任务，整个系统任意节点出现故障的概率远高于单一节点的故障概率，因此高效的智算中心网络故障管理机制是保障智算中心网络与业务性能的关键。这就需要有高效的网络监控模块、故障定位模块及故障恢复模块共同实现智算中心网络的智算管理功

能。这些模块需要根据端到端连通性、网络延迟、网卡与交换机吞吐量及利用率、主机内延迟与主机内带宽、主机侧与交换机侧日志等信息准确定位故障是位于主机间还是主机内，并给出相应的故障处理建议。

对主机间网络故障进行检测从数据中心的发展伊始就是非常重要的一项技术，时至今日已经成为了智算中心中的一种最基本的故障管理技术。Guo等人[34]提出的智算中心网络监控及故障定位解决方案Pingmesh基于端到端服务器之间的全连接相互探测，并以端到端连通性以及RTT为指标对数据中心网络的服务水平协议进行评估；当业务性能下降时，通过观察Pingmesh是否存在指标劣化，可以帮助网络运维人员快速判断问题是否发生在网络路径上；进一步，当网络出现问题时，可以通过对异常五元组进行Traceroute路径追踪，推断出异常的网络设备。Pingmesh能够解决挑战5.1和挑战5.2，但是随着网络规模的逐步增大，其产生的开销也将呈平方增长，在未来应对更大规模的智算中心网络将可能存在极大的瓶颈。

Tan等人[98]提出的NetBouncer方案旨在定位数据中心网络中的非瞬时丢包网络事件，利用了交换机IP-in-IP的新特性技术，可以在源端指定探测包的探测路径，通过探测目标交换机并获取数据包“远路反弹”的结果，NetBouncer可以持续地探测指定的任意数据中心网络路径，并计算出该路径的丢包率。NetBouncer相比于Pingmesh采用了更小的开销来解决挑战5.1中遇到的问题，并在解决挑战5.3中的保障业务服务质量上也做出了一定尝试。

Zhu等人[99]提出的EverFlow方案可以帮助运维人员捕获并分析数据包粒度的流量数据，通过在数据中心交换机上设置特定的匹配规则，并将满足匹配规则的数据包镜像到指定的分析服务器上，实现在数据中心内数据包粒度的网络遥测。EverFlow相比于Pingmesh能够获取并积累更多数据，因而在解决挑战5.2上具有更大的优势，同时也对解决挑战5.3里面的保障业务服务质量做出了一定贡献。

由传统数据中心迈入智算时代后，集群中的网络拓扑愈发复杂，对网络带宽的需求也不断增加，这就造成了智算中心的网络故障类型和数量不断增多。与传统的数据中心相比，智算中心具有更高的网络带宽，并且还在高速持续增长，主机内的延迟和故障相比于网络通信已经不可忽略，这就要求故障管理方案能够有效区分故障类型来自于主机内还是主机间。Liu等人[35]提出的Hostping方案旨在解决主机内网络瓶颈的监测和定位问题，通过环回测试测量主机内各条路径的延迟和带宽，以评估主机内网络状态。Hostping基于二元网络断层扫描算法在主机内网络状态异常的服务器内推断主机内瓶颈链路，并诊断其瓶颈根因；同时还能够对已知的主机内瓶颈实现分钟级响应及瓶颈根因推断，并辅助运维人员识别了多种新的主机内瓶颈。Hostping通过迅速诊断并解决主机内瓶颈根因，实现了高效的智算中心主机内的故障检测，对于在智算中心主机内解决挑战5.1~5.3有较为出色的效果。

在大规模智算中心主机间网络中，还有Liu等人[100]提出的R-Pingmesh方案对Pingmesh做出扩展和升级，改变了传统网络诊断服务的工作流程，在“诊断-基准测试”的基础上增加了智算中心网络集群监控和服务跟踪的功能。R-Pingmesh的核心特性有4个方面：一是增加对RoCE网络特有故障的检测；二是设计了网卡和网络数据包丢失的区分算法，以便对网络故障准确定位；三是提供准确测量端侧主机处理延迟和服务网络RTT的能力，以检测和定位性能瓶颈；四是实现了对服务产生影响的评估，以便为每个问题确定解决方案。R-Pingmesh会根据集群的不同功能来优化全连接探测范围，减少了探测开销，并增加了对不同故障类型的分辨和识别能力，还具有了服务保障措施，对于在智算中心主机间解决挑战5.1~5.3有较为出色的效果。

Guo等人[101]提出的ByteTracker方案考虑到现有方案在运行应用日益复杂的智算中心网络服务器上使用端侧代理容易对业务产生负面影响，在端侧代理故障时难以及时终止代理运行，部署与更新维护方面的运维工作量也颇为繁重。ByteTracker方案采取了由服务器系统内核自动回复RST的探测触发机制，无需端侧部署代理，从而减少了故障管理系统的组件构成，降低了智算中心网络故障管理系统的复杂程度与运维难度。方案仅需在网络中由少量管理服务器集中式发起探测探针，并通过直接发送探测报文覆盖智算中心网络南北向链路，通过IP-in-IP技术覆盖智算中心网络东西向链路。该方案在部署难度和成本极大降低的前提下，更加有效地发现并解决了微小丢包事件、静默比特翻转、芯片级转发错误等故障问题，同时具有高准确率的快速路径追踪能力。ByteTracker能够在智算中心主机间较为出色地解决挑战5.1~5.3，并具有相比Pingmesh更加出色的性能表现和更低的成本需求，但并不能解决R-

Pingmesh 能够探测到的部分 RoCE 网络特定问题。

4 趋势与展望

1)大规模智算中心构建的通用方案设计

随着深度学习参数量呈指数级增长，各种新兴的大模型应用，如深度求索的 DeepSeek[47,102]、OpenAI 的 GPT、Meta 人工智能大型语言模型(large language model Meta AI, LLaMA)[103] 的计算需求指数级增加，承载这些业务的智算中心规模也与日俱增。然而，由于以往没有大规模智算中心的构建先例，不同企业使用的智算中心网络和大模型之间具有高度耦合性，一旦更换部署环境，模型的训练效果和推理能力都将会有显著的改变。因此，如何构建适用于大模型应用的高性能智算系统软硬件通用架构成为学术界与工业界高度关注的问题，其中包含强大的存储与计算架构、跨服务器的多 GPU 可扩展节点、高速 RDMA 通信技术、面向智算中心网络的拥塞控制技术、适用多场景的专用通信库等多个关键技术点。以英伟达设计的 DGX SuperPOD 系统[50] 为例，其计算、存储结构采用完整胖树网络[51] 互联，适用于 Vera Rubin 的节点内最新互联技术：第 6 代 NVLink 支持 28.8 Tbps 带宽[19]，适用于单机 8 卡的节点间 NVLink 6 Switch 支持 230.4 Tbps 的聚合带宽[19]，并可在 NVIDIA Vera Rubin NVL72 平台中提供高达 2 080 Tbps 的聚合 GPU 带宽，NVIDIA UFM[104] 支持横向扩展 InfiniBand 数据中心，传输层采用逐包均衡策略，应用层设计专用通信库 NCCL 加速深度学习训练[26]、UCX(unified communication X)[105] 驱动高速 RDMA 互联。该系统能够提供超大规模的计算能力，从而推动业务创新，为其提供基础设施支持。未来，将会有更多与 NVIDIA DGX SuperPOD 系统类似的智算中心网络完整解决方案，助力大模型的快速部署和推广。

2)适应大模型训练的智算中心网络设计

专门为训练大模型而构建的智算中心网络，是聚焦大模型超大规模参数训练需求的专属网络基础设施，以支撑万卡乃至十万卡级 GPU 集群高效协同为核心目标。智算中心网络围绕大模型分布式并行训练(数据并行/张量并行/流水线并行/混合专家并行等混合并行)产生的 on-off 周期性流量、海量数据同步需求，以“极致带宽、超低时延、低抖动、高可靠”为关键性能指标，通过网络-计算-调度深度协同(如细粒度流量感知调度、端网协同拥塞控制)、开放解耦的规模化组网架构(如多 POD 矩阵设计、HPN 多轨道网络设计)、智能负载均衡策略，解决超大规模集群下的链路竞争、路径拥堵与算力浪费问题，在保障网络稳定性与可扩展性的同时最大化 GPU 利用率，为大模型从千亿到万亿参数级的训练任务提供高效、弹性、可扩展的网络支撑，核心服务于大模型训练这一核心智算中心场景。为此，需要更多地考虑到智算中心网络的流量特征和应用场景，设计出既具有针对性优化能力，又能够具有较强兼容性和扩展性的智算中心网络设计方案。

3)光电协同交换网络的融合演进

随着智算中心计算规模的指数级扩展，传统纯电互联网络为了满足带宽需求采用了复杂的层次化组网结构和巨大的能源消耗，但仍然不能够满足智算中心的扩展需求，目前最高转发速率已经达到 102.4 Tbps[106] 的最先进商用电交换芯片已在带宽密度、能耗效率与端到端时延等方面逐渐逼近物理极限。光电协同交换网络是智算中心网络演进的重要方向，通过在网络架构中引入光交换矩阵与可重构光互连技术，实现了在不同层级间的高带宽、低功耗互联。该架构在短距互联和密集短流传输中采用电交换以实现高灵活性与低时延控制，在长距互联、跨机柜通信与稳定长流传输中采用光交换以获得高带宽与低能耗优势，从而实现“电为主控、光为承载”的协同交换机制[106]。架构在光层采用光电路交换(optical circuit switching, OCS)、阵列波导光栅路由器(arrayed waveguide grating router, AWGR)、自适应波长选择开关(adaptive wavelength selective switch, AWSS)等技术构建无阻塞底座，承载 all-reduce 或 all-to-all 等稳定大流量，实现纳秒级重构与低时延；在电层以 SDN/IP 交换配合 RoCEv2 协议处理突发流量，配合集合通信库优化与无损流控，实现灵活适配；在上层由集中控制器统一编排流量，实现跨层次、跨协议栈的联合优化，支持不同任务间的算力与带宽动态编排。该架构能够有效缓解大规模智算中心中因横向扩展导致的拓扑复杂度与布线压力问题，同时通过光域的快速重构能力，与硅光互连、可编程光交换矩阵、光子芯片网络接口等技术共同演进，形成智算中心的新型底层互联基础设施，为面向大模型与通用人工智能的超大规模智算中心提供高能效比、可扩展的网络支撑。

4)低熵流量特性的充分开发利用

现有的智算中心网络中混合运行了确定性较强的低熵训练数据流量和确定性一般的控制、训练流量，前者吞吐量高、并发多、数据量大但确定性好，

后者吞吐量小、突发多、实时性要求更高且具有一定不确定性。现有的解决方案为了保持网络的通用性,通常将这些流量混合处理,部分解决方案还对这些流量有一定区分能力。进一步提升智算中心网络的性能,需要对这些流量加以明确的区分,根据不同的特点制定专用的处理规则。同时,还应该通过更多有效的技术手段对训练数据流量进行充分调度,从而在不影响训练进度的情况下实现这些大规模流量的错峰传输。这要求智算中心网络协议栈具有跨层级的全方位协同能力,在软硬件深度结合的基础上开发具有普适性和通用性的解决方案。在 3.2 节和 3.3 节所述的 MLT[84] 方案充分考虑了上述要求,设计了面向端到端整体网络传输过程的智算中心网络优化方案,在传输控制、负载均衡等多方面进行协同设计,具有了跨层级的全方位协同能力。

5)多租户智算中心构建

多租户智算中心应用场景中,智算中心负载来自不同物理空间上的多个分布式深度神经网络训练任务。其中单个任务对于 GPU 的算力要求相比于大模型而言要小的多,但是其多任务的特点要求智算中心进行算力虚拟化处理,从而公平、按优先级高效分配给多租户进行深度学习训练。大模型训练通常分为多个阶段,如数据预处理、模型训练、模型评估等,不同阶段对网络的需求有所不同。例如,数据预处理阶段可能需要大量的带宽来传输数据,可将其设定为较高优先级;模型训练阶段对网络延迟较为敏感,应设定为最高优先级;模型评估阶段相对来说对带宽和延迟的要求稍低,可设为中优先级。对于大模型推理,由于其通常需要实时响应,应将其与模型训练阶段同等对待。为了解决其中的流量争用问题,现阶段理想的解决方案是利用网络中的可编程交换机实现动态、尽力的分布式聚合,以便实现最佳资源使用以及用户的公平资源分配[56]。这需要在端主机和交换机资源分配方面进行关键创新,同时思考具备拥塞控制与可靠保障机制的网络堆栈设计。

6)通算智算一体化

当前,算力基础设施正由通用算力为主向通算、智算一体化演进。建设高质量的智算基础设施需要芯片、服务器、网络及数据中心产业紧密协同推进。随着 RDMA 技术的成熟,大规模 GPU 集群训练过程中并不大量使用 CPU 或网络带宽。如何充分调度协同 GPU、CPU 的计算能力是未来实现通算智算一体化的关键问题[107]。这需要与实际应用结合起来,动态感知分析实际应用的流量特性,制定动态的训练策略[108-109]。通算智算一体化的实现将能够在集合通信、传输控制、负载均衡等核心技术层面同时兼顾智算网络和通算网络的特征,既能够确保智算网络高效运行,又能够兼容传统通算网络的稳定运行,实现数据中心网络的进一步进化。

5 总 结

智算中心是面向未来的数据中心形态,具有广阔的发展空间。但智算中心网络跨越式发展的特点,决定了在传统数据中心中的很多技术需要进行适应性变革和改变才能够被加以利用。智算中心以周期性通信交互为主的流量特征给智算中心网络的发展带来了巨大的挑战,同时也迎来了全新的机遇。利用好这种流量特征改进智算中心网络的整体结构及关键技术,成为了目前主要的研究方向。这些关键技术包括实现数据通信传输的集合通信机制、减少网络拥塞的传输控制方案、提升网络利用效率和带宽的负载均衡算法、保障无损网络的链路控制方案以及确保智算中心持续稳定运行的故障管理机制等。在未来,智算中心也将以这些技术为基础,通过一体化分层专用方案设计,实现更大规模的智算中心,同时为 DeepSeek, ChatGPT 等生成式大模型和其他多租户应用场景进行服务,并最终实现集通算、智算、超算一体化的高性能智算中心。

作者贡献声明:张娇设计论文的总体框架及核心思路;刘怿睿负责论文传输控制、链路控制和故障管理相关内容;康志远负责论文集合通信部分内容;刘军卫、孟令坤、徐军、赵兴华和季忠铭共同负责论文负载均衡相关内容和集合通信部分内容;黄韬提出整体指导意见。

参 考 文 献

[1] CAICT. White paper on China comprehensive computing power evaluation (2023)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202309/P020230906485351670771.pdf (in Chinese)
(中国信息通信研究院. 中国综合算力评价白皮书(2023 年)[EB/OL]. 北京: 中国信息通信研究院, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202309/P020230906485351670771.pdf)

[2] CAICT. White paper on China's computing power development index (2023)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202309/P020230914584614752938.

pdf (in Chinese)
（中国信息通信研究院. 中国算力发展指数白皮书(2023 年)[EB/OL]. 北京：中国信息通信研究院，2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202309/P020230914584614752938.pdf）

[3] CAICT. Sustainable information disclosure development report for the Internet industry (2023)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202311/P020231129695427339720.pdf (in Chinese)
（中国信息通信研究院. 互联网行业可持续信息披露发展报告(2023 年)[EB/OL]. 北京：中国信息通信研究院，2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202311/P020231129695427339720.pdf）

[4] CAICT. White paper on global digital governance (2023)[EB/OL]. Beijing: CAICT, 2024[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202401/P020240103389490640356.pdf (in Chinese)
（中国信息通信研究院. 全球数字治理白皮书(2023 年)[EB/OL]. 北京：中国信息通信研究院，2024[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202401/P020240103389490640356.pdf）

[5] CAICT. White paper on global digital economy (2023)[EB/OL]. Beijing: CAICT, 2024[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202401/P020240109492552259509.pdf (in Chinese)
（中国信息通信研究院. 全球数字经济白皮书(2023 年)[EB/OL]. 北京：中国信息通信研究院，2024[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202401/P020240109492552259509.pdf）

[6] CAICT. Research report on the development of China's digital economy (2023)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202304/P020230427572038320317.pdf (in Chinese)
（中国信息通信研究院. 中国数字经济发展研究报告(2023 年)[EB/OL]. 北京：中国信息通信研究院，2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202304/P020230427572038320317.pdf）

[7] CAICT. White paper on cloud computing (2023)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202307/P020230725521473129120.pdf (in Chinese)
（中国信息通信研究院. 云计算白皮书(2023 年)[EB/OL]. 北京：中国信息通信研究院，2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202307/P020230725521473129120.pdf）

[8] CAICT. White paper on data elements (2023)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202309/P020231103487803108185.pdf (in Chinese)
（中国信息通信研究院. 数据要素白皮书(2023 年)[EB/OL]. 北京：中国信息通信研究院，2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202309/P020231103487803108185.pdf）

[9] CAICT. White paper on big data (2022)[EB/OL]. Beijing: CAICT, 2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202301/P020230104388100740258.pdf (in Chinese)
（中国信息通信研究院. 大数据白皮书(2022 年)[EB/OL]. 北京：中国信息通信研究院，2023[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202301/P020230104388100740258.pdf）

[10] CAICT. White paper on artificial intelligence (2022)[EB/OL]. Beijing: CAICT, 2022[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202204/P020220412613255124271.pdf (in Chinese)
（中国信息通信研究院. 人工智能白皮书(2022 年)[EB/OL]. 北京：中国信息通信研究院，2022[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202204/P020220412613255124271.pdf）

[11] CAICT. White paper on AI framework development (2022)[EB/OL]. Beijing: CAICT, 2022[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202202/P020220226369908606520.pdf (in Chinese)
（中国信息通信研究院. AI 框架发展白皮书(2022 年)[EB/OL]. 北京：中国信息通信研究院，2022[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202202/P020220226369908606520.pdf）

[12] China Mobile. White paper on new generation artificial intelligence data center network technology[EB/OL]. Beijing: China Mobile, 2022[2024-01-24]. https://cmri.chinamobile.com/wp-content/uploads/2022/12/%E6%96%B0%E4%B8%80%E4%BB%A3%E6%99%BA%E7%AE%97%E4%B8%AD%E5%BF%83%E7%BD%91%E7%BB%9C%E6%8A%80%E6%9C%AF%E7%99%BD%E7%9A%AE%E4%B9%A6.pdf (in Chinese)
（中国移动. 新一代智算中心网络技术白皮书(2022 年)[EB/OL]. 北京：中国移动，2022[2024-01-24]. https://cmri.chinamobile.com/wp-content/uploads/2022/12/%E6%96%B0%E4%B8%80%E4%BB%A3%E6%99%BA%E7%AE%97%E4%B8%AD%E5%BF%83%E7%BD%91%E7%BB%9C%E6%8A%80%E6%9C%AF%E7%99%BD%E7%9A%AE%E4%B9%A6.pdf）

[13] China Telecom Research Institute, Tianyi Zhiku. White paper on development of artificial intelligence data center industry (2023)[EB/OL]. Beijing: China Telecom Research Institute, 2023[2024-01-24]. http://www.ecconsortium.org/Uploads/file/20231023/1698032861186142.pdf (in Chinese)
（中国电信研究院，天翼智库. 智算产业发展白皮书(2023 年)[EB/OL]. 北京：中国电信研究院，2023[2024-01-24]. http://www.ecconsortium.org/Uploads/file/20231023/1698032861186142.pdf）

[14] China Mobile Research Institute. White paper on development of artificial intelligence data center for AI large language model (2023)[EB/OL]. Beijing: China Mobile Research Institute, 2023[2024-01-24]. http://www.ecconsortium.org/Uploads/file/20230517/1684313521798632.pdf (in Chinese)
（中国移动通信研究院. 面向 AI 大模型的智算中心网络演进白皮书(2023 年)[EB/OL]. 北京：中国移动通信研究院，2023[2024-01-24]. http://www.ecconsortium.org/Uploads/file/20230517/1684313521798632.pdf）

[15] H3C Group, CAICT. White paper on development of intelligence computing power (2023)[EB/OL]. Hangzhou: H3C Group, 2023[2024-01-24]. http://www.ecconsortium.org/Uploads/file/20230830/1693375872266388.pdf (in Chinese)
（新华三集团，中国信息通信研究院. 2023 智能算力发展白皮书[EB/OL]. 杭州：新华三集团，2023[2024-01-24]. http://www.ecconsortium.org/Uploads/file/20230830/1693375872266388.pdf）

[16] Brodtkorb A R, Hagen T R, Sætra M L. Graphics processing unit (GPU) programming strategies and trends in GPU computing[J]. Journal of Parallel and Distributed Computing, 2013, 73(1): 4−13

[17] CAICT. White paper on AI-generated content (AIGC) (2022)[EB/OL]. Beijing: CAICT, 2022[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202209/P020220902534520798735.pdf (in Chinese)
（中国信息通信研究院. 人工智能生成内容(AIGC)白皮书(2022 年)[EB/OL]. 北京：中国信息通信研究院，2022[2024-01-24]. http://www.caict.ac.cn/kxyj/qwfb/bps/202209/P020220902534 5

20798735.pdf)
[18] NVIDIA. DGX A100: Universal system for AI infrastructure[EB/OL]. [2024-01-24]. https://www.nvidia.com/en-us/data-center/dgx-a100/
[19] NVIDIA. NVLink & NVSwitch: Fastest HPC data center platform [EB/OL]. [2024-01-24]. https://www.nvidia.com/en-us/data-center/nvlink/
[20] Weingram A, Li Yuke, Qi Hao, et al. xCCL: A survey of industry-led collective communication libraries for deep learning[J]. Journal of Computer Science and Technology, 2023, 38(1): 166−195
[21] Krizhevsky A. One weird trick for parallelizing convolutional neural networks[J]. arXiv preprint, arXiv: 1404.5997, 2014
[22] Huang Yanping, Cheng Youlong, Bapna A, et al. GPipe: Efficient training of giant neural networks using pipeline parallelism[J]. Advances in Neural Information Processing Systems, 2019, 32: 620−629
[23] Shoeybi M, Patwary M, Puri R, et al. Megatron-LM: Training multi-billion parameter language models using model parallelism[J]. arXiv preprint, arXiv: 1909.08053, 2019
[24] Aminabadi R Y, Rajbhandari S, Awan A A, et al. DeepSpeed-inference: Enabling efficient inference of Transformer models at unprecedented scale[C]//Proc of the 35th Int Conf for High Performance Computing, Networking, Storage and Analysis. Piscataway, NJ: IEEE, 2022: 1−15
[25] Xu Jing, Wang Zhan, Yuan Guojun, et al. Survey on fine-grained flow control for artificial intelligence data centers[J]. Journal of Computer Research and Development, 2025, 62(11): 2806−2825 (in Chinese)
(许晶, 王展, 元国军, 等. 面向智算中心的细粒度流量控制技术综述[J]. 计算机研究与发展, 2025, 62(11): 2806−2825)
[26] NVIDIA. Nvidia NCCL[EB/OL]. (2023-11-20)[2024-01-24]. https://github.com/nvidia/nccl
[27] Cai Zixian, Liu Zhengyang, Maleki S, et al. Synthesizing optimal collective algorithms[C]//Proc of the 26th ACM SIGPLAN Symp on Principles and Practice of Parallel Programming. New York: ACM, 2021: 62−75
[28] Shah A, Chidambaram V, Cowan M, et al. TACCL: Guiding collective algorithm synthesis using communication sketches[C]//Proc of the 20th USENIX Symp on Networked Systems Design and Implementation (NSDI 23). Berkeley, CA: USENIX Association, 2023: 593−612
[29] Liu Xuting, Arzani B, Kakarla S K R, et al. Rethinking machine learning collective communication as a multi-commodity flow problem[C]//Proc of the 38th ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 16−37
[30] Mittal R, Lam V T, Dukkipati N, et al. TIMELY: RTT-based congestion control for the datacenter[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 537−550
[31] Kumar G, Dukkipati N, Jang K, et al. Swift: Delay is simple and effective for congestion control in the datacenter[C]//Proc of the 34th Annual Conf of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2020). New York: ACM, 2020: 514−528
[32] Wang Weitao, Moshref M, Li Yuliang, et al. Poseidon: Efficient, robust, and practical datacenter CC via deployable INT[C]//Proc of the 20th USENIX Symp on Networked Systems Design and Implementation (NSDI 23). Berkeley, CA: USENIX Association, 2023: 255−274
[33] NVIDIA. Nvidia InfiniBand[EB/OL]. [2024-01-24]. https://www.nvidia.com/en-us/networking/infiniband-adapters/
[34] Guo Chuanxiong, Yuan Lihua, Xiang Dong, et al. Pingmesh: A large-scale system for data center network latency measurement and analysis[C]//Proc of the 29th ACM Conf on Special Interest Group on Data Communication (SIGCOMM 2015). New York: ACM, 2015: 139−152
[35] Liu Kefei, Jiang Zhuo, Zhang Jiao, et al. Hostping: Diagnosing intra-host network bottlenecks in RDMA servers[C]//Proc of the 20th USENIX Symp on Networked Systems Design and Implementation (NSDI 23). Berkeley, CA: USENIX Association, 2023: 15−29
[36] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877−1901
[37] Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report[J]. arXiv preprint, arXiv: 2303.08774, 2023
[38] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint, arXiv: 1802.05365, 2023
[39] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint, arXiv: 1810.04805, 2018
[40] Solaiman I, Brundage M, Clark J, et al. Release strategies and the social impacts of language models[J]. arXiv preprint, arXiv: 1908.09203, 2019
[41] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485−5551
[42] Rosset C. Turing-NLG: A 17-billion-parameter language model by Microsoft[EB/OL]. (2020-02-13)[2024-01-24]. https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/
[43] Smith S, Patwary M, Norick B, et al. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530b, a large-scale generative language model[J]. arXiv preprint, arXiv: 2201.11990, 2022
[44] Chowdhery A, Narang S, Devlin J, et al. PaLM: Scaling language modeling with pathways[J]. Journal of Machine Learning Research, 2023, 24(240): 1−113
[45] Anthropic. Introducing the next generation of Claude[EB/OL]. (2024-03-04)[2024-10-24]. https://www.anthropic.com/news/claude-3-family
[46] Grattafiori A, Dubey A, Jauhri A, et al. The LLaMA 3 herd of models[J]. arXiv preprint, arXiv: 2407.21783, 2024
[47] Guo Daya, Yang Dejian, Zhang Haowei, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning[J]. arXiv preprint, arXiv: 2501.12948, 2025
[48] Abacha A B, Yim W, Fu Yujuan, et al. Medec: A benchmark for medical error detection and correction in clinical notes[J]. arXiv preprint, arXiv:2412.19260, 2024

[49] Qian Kun, Xi Yongqiang, Cao Jiamin, et al. Alibaba HPN: A data center network for large language model training[C]//Proc of the 38th ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 691−706

[50] Choquette J, Gandhi W. Nvidia A100 GPU: Performance & innovation for GPU computing[C]//Proc of the 32nd IEEE Hot Chips Symp (HCS). Los Alamitos, CA: IEEE Computer Society, 2020: 1−43

[51] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 63−74

[52] Dai Damai, Deng Chengqi, Zhao Chenggang, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models[J]. arXiv preprint, arXiv: 2401.06066, 2024

[53] Potluri S, Hamidouche K, Venkatesh A, et al. Efficient inter-node MPI communication using GPUDirect RDMA for InfiniBand clusters with NVIDIA GPUs[C]//Proc of the 42nd Int Conf on Parallel Processing. Piscataway, NJ: IEEE, 2013: 80−89

[54] Zhai Mingshu, He Jiaao, Ma Zixuan, et al. SmartMoE: Efficiently training sparsely-activated models through combining offline and online parallelization[C]//Proc of the 32nd USENIX Annual Technical Conf (USENIX ATC 23). Berkeley, CA: USENIX Association, 2023: 961−975

[55] Shi Haibin, Zhu Guangxi, Zhu Peng, et al. Enhanced on/off dynamic traffic model and its simulation[J]. Computer Science, 2009, 36(2): 75−77 (in Chinese)
(史海滨, 朱光喜, 祝鹏, 等. 一种增强的 ON/OFF 瞬时流量模型及其仿真[J]. 计算机科学, 2009, 36(2): 75−77)

[56] Sapio A, Canini M, Ho C Y, et al. Scaling distributed machine learning with in-network aggregation[C]//Proc of the 18th USENIX Symp on Networked Systems Design and Implementation (NSDI 21). Berkeley, CA: USENIX Association, 2021: 785−808

[57] Gebara N, Ghobadi M, Costa P. In-network aggregation for shared machine learning clusters[J]. Proceedings of Machine Learning and Systems, 2021, 3: 829−844

[58] Lao C L, Le Yanfang, Mahajan K, et al. ATP: In-network aggregation for multi-tenant learning[C]//Proc of the 18th USENIX Symp on Networked Systems Design and Implementation (NSDI 21). Berkeley, CA: USENIX Association, 2021: 741−761

[59] Luo Liang, Nelson J, Ceze L, et al. Parameter hub: A rack-scale parameter server for distributed deep neural network training[C]// Proc of the 8th ACM Symp on Cloud Computing. New York: ACM, 2018: 41−54

[60] Deng Wei, Pan Junwei, Zhou Tian, et al. DeepLight: Deep lightweight feature interactions for accelerating CTR predictions in ad serving[C]//Proc of the 14th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2021: 922−930

[61] Li Shen, Zhao Yanli, Varma R, et al. PyTorch distributed: Experiences on accelerating data parallel training[J]. Proceedings of the VLDB Endowment, 2020, 13(12): 3005−3018

[62] Mahajan K, Chu C H, Sridharan S, et al. Better together: Jointly optimizing ML collective scheduling and execution planning using SYNDICATE[C]//Proc of the 20th USENIX Symp on Networked Systems Design and Implementation (NSDI 23). Berkeley, CA: USENIX Association, 2023: 809−824

[63] Li Ang, Song S L, Chen Jieyang, et al. Evaluating modern GPU interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 31(1): 94−110

[64] Graham R L, Shipman G M, Barrett B W, et al. Open MPI: A high-performance, heterogeneous MPI[C]//Proc of the 8th IEEE Int Conf on Cluster Computing. Piscataway, NJ: IEEE, 2006: 1−9

[65] Sanders J, Kandrot E. CUDA by Example: An Introduction to General-purpose GPU Programming[M]. Sydney, Australia: Addison-Wesley Professional, 2010

[66] Wang Guanhua, Venkataraman S, Phanishayee A, et al. Blink: Fast and generic collectives for distributed ML[J]. Proceedings of Machine Learning and Systems, 2020, 2: 172−186

[67] Neugebauer R, Antichi G, Zazo J F, et al. Understanding PCIe performance for end host networking[C]//Proc of the 32nd Conf of the ACM Special Interest Group on Data Communication (SIGCOMM 2018). New York: ACM, 2018: 327−341

[68] Foley D, Danskin J. Ultra-performance Pascal GPU and NVLink interconnect[J]. IEEE Micro, 2017, 37(2): 7−17

[69] Alibaba Cloud. Alibaba ACCL[EB/OL]. (2024-01-04)[2024-01-24]. https://help.aliyun.com/zh/pai/user-guide/installation-1 (in Chinese)
(阿里云. 阿里 ACCL[EB/OL]. (2024-01-04)[2024-01-24]. https://help.aliyun.com/zh/pai/user-guide/installation-1)

[70] Dong Jianbo, Wang Shaochuang, Feng Fei, et al. Accl: Architecting highly scalable distributed training systems with highly efficient collective communication library[J]. IEEE Micro, 2021, 41(5): 85−92

[71] Tencent Cloud. Tencent TCCL[EB/OL]. (2023-09-18)[2024-01-24]. https://cloud.tencent.com/document/product/1646/93319 (in Chinese)
(腾讯云. 腾讯 TCCL[EB/OL]. (2023-09-18)[2024-01-24]. https://cloud.tencent.com/document/product/1646/93319)

[72] Microsoft. MSCCL: Microsoft Collective Communication Library [EB/OL]. (2023-09-21)[2024-01-24]. https://github.com/microsoft/msccl/

[73] Cao Jiamin, Guan Yi, Qian Kun, et al. Crux: GPU-efficient communication scheduling for deep learning training[C]//Proc of the38thACM SIGCOMM 2024 Conf. New York: ACM, 2024: 1−15

[74] Rajasekaran S, Ghobadi M, Akella A. CASSINI: Network-aware job scheduling in machine learning clusters[C]//Proc of the 21st USENIX Symp on Networked Systems Design and Implementation (NSDI 24). Berkeley, CA: USENIX Association, 2024: 1403−1420

[75] Huang Shan, Dong Dezun, Bai Wei. Congestion control in high-speed lossless data center networks: A survey[J]. Future Generation Computer Systems, 2018, 89: 360−374

[76] Wu Hangxing, Yang Xiaolong, Zhang Min. Congestion control in data center networks: A survey and new perspectives[J]. Applied Mechanics and Materials, 2014, 462: 1028−1035

[77] Zhang Jiao, Ren Fengyuan, Lin Chuang. Survey on transport control in data center networks[J]. IEEE Network, 2013, 27(4): 22−26

[78] Zhu Yibo, Eran H, Firestone D, et al. Congestion control for large-scale RDMA deployments[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 523−536

[79] Li Yuliang, Miao Rui, Liu H H, et al. HPCC: High precision congestion control[C]//Proc of the 33rd ACM Special Interest Group

on Data Communication (SIGCOMM 2019). New York: ACM, 2019: 44−58

[80] Cho I, Jang K, Han D. Credit-scheduled delay-bounded congestion control for datacenters[C]//Proc of the 31st Conf of the ACM Special Interest Group on Data Communication (SIGCOMM 2017). New York: ACM, 2017: 239−252

[81] Montazeri B, Li Yilong, Alizadeh M, et al. Homa: A receiver-driven low-latency transport protocol using network priorities[C]//Proc of the 32nd Conf of the ACM Special Interest Group on Data Communication (SIGCOMM 2018). New York: ACM, 2018: 221−235

[82] Hu Shuihai, Bai Wei, Zeng Gaoxiong, et al. Aeolus: A building block for proactive transport in datacenters[C]//Proc of the 34th Annual Conf of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2020). New York: ACM, 2020: 422−434

[83] Rajasekaran S, Narang S, Zabreyko A A, et al. MLTCP: A distributed technique to approximate centralized flow scheduling for machine learning[C]//Proc of the 23rd ACM Workshop on Hot Topics in Networks. New York: ACM, 2024: 167−176

[84] Wang Hao, Tian Han, Chen Jingrong, et al. Towards domain-specific network transport for distributed DNN training[C]//Proc of the 21st USENIX Symp on Networked Systems Design and Implementation (NSDI 24). Berkeley, CA: USENIX Association, 2024: 1421−1443

[85] Lu Yifei, Li Jingqi, Li Shuren, et al. A fine-grained packet loss tolerance transmission algorithm for communication optimization in distributed deep learning[J]. IEEE Transactions on Network and Service Management, 2024, 21(6): 6112−6125

[86] Zhao Xiaoyang, Wu Chuan, Zhu Xia. Dynamic flow scheduling for DNN training workloads in data centers[J]. IEEE Transactions on Network and Service Management, 2024, 21(6): 6643−6657

[87] Mittal R, Shpiner A, Panda A, et al. Revisiting network support for RDMA[C]//Proc of the 32nd Conf of the ACM Special Interest Group on Data Communication (SIGCOMM 2018). New York: ACM, 2018: 313−326

[88] Huang Peihao, Chen Guo, Zhang Xin, et al. Fast and scalable selective retransmission for RDMA[C]//Proc of the 44th IEEE Conf on Computer Communications (INFOCOM 2025). Piscataway, NJ: IEEE, 2025: 1−10

[89] Zhang Jiao, Yu F R, Wang Shuo, et al. Load balancing in data center networks: A survey[J]. IEEE Communications Surveys & Tutorials, 2018, 20(3): 2324−2352

[90] Thaler D, Hopps C. MultipathIssues in Unicast and Multicast Next-hop Selection[S/OL]. 2000[2024-01-24]. https://doi.org/10.17487/RFC2991

[91] Chen Guo, Lu Yuanwei, Li Bojie, et al. MP-RDMA: Enabling RDMA with multi-path transport in datacenters[J]. IEEE/ACM Transactions on Networking, 2019, 27(6): 2308−2323

[92] Song C H, Khooi X Z, Joshi R, et al. Network load balancing with in-network reordering support for RDMA[C]//Proc of the 37th ACM SIGCOMM 2023 Conf. New York: ACM, 2023: 816−831

[93] NVIDIA. NVIDIA Spectrum-X: Ethernet networking platform for AI[EB/OL]. [2024-01-24]. https://www.nvidia.com/en-us/networking/spectrumx/

[94] Alizadeh M, Edsall T, Dharmapurikar S, et al. CONGA: Distributed congestion-aware load balancing for datacenters[C]//Proc of the 28th ACM Conf on SIGCOMM. New York: ACM, 2014: 503−514

[95] Ghorbani S, Yang Zibin, Godfrey P B, et al. Drill: Micro load balancing for low-latency data center networks[C]//Proc of the 31st Conf of the ACM Special Interest Group on Data Communication (SIGCOMM 2017). New York: ACM, 2017: 225−238

[96] Qureshi M A, Cheng Y, Yin Qianwen, et al. PLB: Congestion signals are simple and effective for network load balancing[C]//Proc of the 36th ACM SIGCOMM 2022 Conf. New York: ACM, 2022: 207−218

[97] Guo Chuanxiong, Wu Haitao, Deng Zhong, et al. RDMA over commodity Ethernet at scale[C]//Proc of the 30th ACM SIGCOMM Conf. New York: ACM, 2016: 202−215

[98] Tan Cheng, Jin Ze, Guo Chuanxiong, et al. NetBouncer: Active device and link failure localization in data center networks[C]//Proc of the 16th USENIX Symp on Networked Systems Design and Implementation (NSDI 19). Berkeley, CA: USENIX Association, 2019: 599−614

[99] Zhu Yibo, Kang Nanxi, Cao Jiaxin, et al. Packet-level telemetry in large datacenter networks[C]//Proc of the 29th ACM Conf on Special Interest Group on Data Communication (SIGCOMM 2015). New York: ACM, 2015: 479−491

[100] Liu Kefei, Jiang Zhuo, Zhang Jiao, et al. R-Pingmesh: A service-aware RoCE network monitoring and diagnostic system[C]//Proc of the 38th ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 554−567

[101] Guo Shixian, Liu Kefei, Lai Yulin, et al. ByteTracker: An agentless and real-time path-aware network probing system[C]//Proc of the 39th ACM SIGCOMM 2025 Conf. New York: ACM, 2025: 541−553

[102] Liu Aixin, Feng Bei, Xue Bing, et al. DeepSeek-v3 technical report[J]. arXiv preprint, arXiv: 2412.19437, 2024

[103] Touvron H, Lavril T, Izacard G, et al. LLaMA: Open and efficient foundation language models[J]. arXiv preprint, arXiv: 2302.13971, 2023

[104] NVIDIA. NVIDIA unified fabric manager (UFM)[EB/OL]. [2024-01-24]. https://www.nvidia.com/en-us/networking/infiniband/ufm/

[105] NVIDIA. NVIDIA HPC-X[EB/OL]. [2024-01-24]. https://developer.nvidia.com/networking/hpc-x

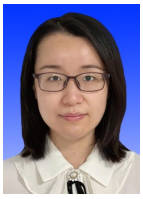
[106] Hunan University, China Unicom Research Institute, China Unicom Software Institute, et al. White paper on full-stack technologies for photonic-electrical co-engineered switching networks in artificial intelligence data center[EB/OL]. Changsha: Hunan University, 2025[2025-12-12]. https://gitee.com/bangboliang/opti-net-lab/raw/master/whitepaper.pdf (in Chinese)
(湖南大学, 中国联通研究院, 中国联通软件研究院, 等. 智算中心光电协同交换网络全栈技术白皮书[EB/OL]. 长沙: 湖南大学, 2025[2025-12-12]. https://gitee.com/bangboliang/opti-net-lab/raw/master/whitepaper.pdf)

[107] Hwang C, Park K S, Shu Ran, et al. ARK: GPU-driven code execution for distributed deep learning[C]//Proc of the 20th USENIX Symp on Networked Systems Design and Implementation (NSDI 23). Berkeley, CA: USENIX Association, 2023: 87−101

[108] Jiang Yimin, Zhu Yibo, Lan Chang, et al. A unified architecture for

accelerating distributed DNN training in heterogeneous GPU/CPU clusters[C]//Proc of the 14th USENIX Symp on Operating Systems Design and Implementation (OSDI 20). Berkeley, CA: USENIX Association, 2020: 463−479

[109] Chen Jingrong, Zhang Hong, Zhang Wei, et al. NetHint: White-box networking for multi-tenant data centers[C]//Proc of the 19th USENIX Symp on Networked Systems Design and Implementation (NSDI 22). Berkeley, CA: USENIX Association, 2022: 1327−1343



Zhang Jiao, born in 1986. PhD, professor, PhD supervisor. Senior member of CCF. Her main research interests include AIDC networking and network transmission control protocols.

张　娇, 1986 年生。博士，教授，博士生导师。CCF 高级会员。主要研究方向为智算中心网络、网络传输控制协议。



Liu Yirui, born in 2001. PhD candidate. His main research interests include AIDC networking, network fault management, and network transmission control protocols.

刘怿睿, 2001 年生。博士研究生。主要研究方向为智算中心网络、网络故障管理、网络传输控制协议。



Kang Zhiyuan, born in 2002. PhD candidate. His main research interests include AIDC networking and collective communications.

康志远, 2002 年生。博士研究生。主要研究方向为智算中心网络、集合通信。



Liu Junwei, born in 1983. Master, engineer, chief expert of China Mobile. His main research interests include cloud computing and computing power networks.

刘军卫, 1983 年生。硕士，工程师，中国移动首席专家。主要研究方向为云计算、算力网络。



Meng Lingkun, born in 1990. Master, engineer. His main research interests include cloud computing infrastructure, edge computing, and intelligent computing.

孟令坤, 1990 年生。硕士，工程师。主要研究方向为云计算基础设施、边缘计算、智能计算。



Xu Jun, born in 1986. PhD, engineer. His main research interests include cloud computing, high performance network, and artificial intelligence.

徐　军, 1986 年生。博士，工程师。主要研究方向为云计算、高性能网络、人工智能。



Zhao Xinghua, born in 1990. PhD, senior engineer. His main research interests include cloud computing infrastructure, high performance network, transmission control protocols, and control theory and control methods.

赵兴华, 1990 年生。博士，高级工程师。主要研究方向为云计算基础设施、高性能网络、传输控制协议、控制理论与控制方法。



Ji Zhongming, born in 1994. PhD candidate. His main research interests include high performance network, artificial intelligence, and computing power network.

季忠铭, 1994 年生。博士研究生。主要研究方向为高性能网络、人工智能、算力网络。



Huang Tao, born in 1980. PhD, professor, PhD supervisor. His main research interests include network architecture, software-defined networks, and content-centric networking.

黄　韬, 1980 年生。博士，教授，博士生导师。主要研究方向为网络体系架构、软件定义网络、内容中心网络。