星脉网络：面向GPU集群集合通信与集中式路由的协同优化

Astral Network: Co-Optimizing Collectivce Communication and Central Traffic Routing for GPU Clusters

李宝嘉/LI Baojia[1]，何春志/HE Chunzhi[1]，
夏寅贲/XIA Yinben[1]，何泽坤/HE Zekun[1]，
王晓亮/WANG Xiaoliang[2]
（1. 腾讯科技（深圳）有限公司，中国 深圳 518000；
2. 南京大学，中国 南京 210023）
(1. Tencent Technology (Shenzhen) Co., Ltd., Shenzhen 518000, China;
2. Nanjing University, Nanjing 210023, China)

DOI：10.12142/ZTETJ.202502002
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20250424.0957.004
网络出版日期：2025-04-24
收稿日期：2025-03-24

摘要：图形处理器（GPU）集群网络流量不断增加，运营难度明显加大，这给高性能大规模GPU集群网络系统的构建带来新的挑战与机遇。提出了一种能够实现超10万GPU集群互联的无损高性能网络方案——星脉网络。GPU集群网络需要联合优化端侧的集合通信库和网络路由控制器，以实现多路径的高效集合通信。为此，针对星脉网络研发了端侧集合通信库（TCCL）以实现最短的跨节点路径规划，同时还开发了全局优化路由器（GOR）以避免路径冲突导致的网络拥塞。在腾讯大模型GPU集群中，星脉网络方案和公开GPU集群方案（NVIDIA NCCL）的对比结果表明：星脉网络可以实现25%的集合通信带宽提升，同时避免80%的由流量冲突造成的网络拥塞问题。

关键词：大规模GPU集群；集合通信；负载均衡

Abstract: The network traffic of the graphics processing unit (GPU) cluster is continuously increasing, and the operation complexity has significantly increased, which brings new challenges and opportunities to the construction of high-performance large-scale GPU cluster network systems. To address this, we propose Astral Network—a lossless high-performance network architecture capable of interconnecting over 100 000 GPUs. GPU-centric networks require joint optimization of the collective communication library at the host and centralized routing controller to achieve efficient collective communication over multiple paths. Therefore, Tencent developed a collective communication library (TCCL) for Astral Network to achieve the shortest path planning across nodes, and a global optimized router (GOR) to avoid network congestion caused by route conflict through traffic planning. In Tencent's large-scale GPU-centric clusters, the comparison results between Astral Network and the publicly available GPU-centric network (i.e., NVIDIA NCCL) show that: Astral Network achieves 25% higher collective communication bandwidth while reducing traffic conflict-induced congestion by 80%.

Keywords: large-scale GPU clusters; collective communication; load balancing

引用格式：李宝嘉, 何春志, 夏寅贲, 等. 星脉网络：面向GPU集群集合通信与集中式路由的协同优化 [J]. 中兴通讯技术, 2025, 31(2): 3-13. DOI: 10.12142/ZTETJ.202502002

Citation: LI B J, HE C Z, XIA Y B, et al. Astral network: co-optimizing collectivce communication and central traffic routing for GPU clusters [J]. ZTE technology journal, 2025, 31(2): 3-13. DOI: 10.12142/ZTETJ.202502002

生成式人工智能（AIGC）的迅猛发展，推动着人工智能（AI）大模型参数量从亿级飙升到万亿级[1]。模型参数规模增长与架构升级对底层网络提出新的要求。一个拥有4 000张图形处理器（GPU）卡的集群，包含超过150个网络设备和1万条路由路径。若要支持1.6万张GPU卡互联，网络设备数量需要超过1 000个，路由表项超过5万条。集群规模越大，所产生的通信损耗就会越高[2]。同时，AI训练的通信模式与传统的通信模式差异较大，不同大模型架构在通信模式上也各有不同。部分大模型训练过程中，通信占比最高可达50%。分布式计算模式意味着，单点故障可能导致整个集群不可用。因此，在故障发生时，需要快速定位问题并恢复训练，将损失降到最低。如何在大规模组网的背景下，提升通信效率，降低通信占比，保障训练的稳定性与高可用性，进而提升GPU利用率和模型训练效率，是AI网络亟待解决的核心问题。本文介绍了腾讯星脉网络应对这些挑战的思路和解决方案。

星脉网络架构设计基于两个原则：可扩展性和高可靠性。可扩展性是指能够根据AI业务训练的需求动态扩展集群规模，保证集群通信性能呈线性提升。可靠性是指保证架构能够承受一定程度的网络设备故障风险，使AI业务训练不出现中断。为符合上述原则，我们采用图1所示的多轨道网络架构。该架构通过同轨道交换机连接每个服务器中具有相同序号的网卡，然后通过二层（Spine）交换机实现一层（Leaf）交换机的全互联，最终形成一个两层的多轨道网络拓扑。具体地，为了实现跨GPU服务器的高速互联，每个服务器均配有8个端口速率为400 Gbit/s且支持基于融合以太网远端内存直接访问技术（RoCE）的网卡（NIC）。每个网卡通过同轨道的Leaf层交换机与其他服务器的同序号网卡实现互联。如图1所示，第一台服务器的NIC1到第二台服务器的NIC1的流量通过Rail 1交换机传输。此外，网卡采用双端口配置，每个端口连接到不同交换机。这可以有效降低端口故障的影响，提升集群互联的可靠性。网卡采用RoCEv2协议，并且使用优先级流量控制（PFC）实现无损网络。为了有效避免流量拥塞，我们采用带有动态水线的数据中心拥塞控制机制（DCQCN），以保证高通信吞吐。

1 LLM训练对GPU集群网络带来的挑战

以大语言模型（LLM）训练为主的GPU集群与传统数据中心或高性能计算场景有显著不同。深入理解和利用这些特性对于优化训练过程至关重要。LLM训练网络的特点主要表现在以下3个方面：

1）网络边界扩展到服务器内。传统网络规划主要关注节点间互联，而GPU集群网络需同时考虑节点内NVLink/NVSwitch与节点之间RoCE连接的异构互联架构。这种扩展的网络边界要求通信算法进行针对性设计：一是，突破传统NIC间传输范式，实现跨节点GPU显存之间的数据交换；二是，针对异构互联架构，开发多路径带宽聚合方案。

2）GPU通信模式呈现稀疏性和周期性。如图2（a）所示，以往大规模RoCE网络设计，基本基于两种假设：一是网络内所有节点可任意互联；二是流量随机且不可预测。图2（b）展示了我们从大模型训练网络中收集的NIC出口流量数据在每个训练迭代后，所呈现出的周期性和可预测性特点。这种结构化的流量模式为规划路由时最大化GPU集群的利用率提供了参考。

3）大模型训练对网络抖动敏感。尽管传统网络在每个链路上承载大量流，但流的数量多且大多数流较小。在这种情况下，链路负载相对均衡。相对而言，GPU网络中的流数量相对较少，但每个流的体积却较大。例如，在一个拥有1 000个NVIDIA A100卡的集群中，每个流的峰值带宽可达180 Gbit/s，而此时流的数量仅为4 000，这容易导致网络拥塞。交换机不能单纯依靠与终端主机拥塞控制的交互来平衡流量，否则可能导致吞吐量下降。因此，我们需要一种有效的流负载均衡策略和动态调度机制，以避免拥塞状况，同时充分挖掘网络多路径的潜力。

为应对上述挑战，我们提出了一种新型流量管理框架。该框架借助集合通信优化和集中式路由规划的协同设计，实现以下关键成效：集合通信时延降低25%，链路带宽利用率提升至90%，网络拥塞发生概率控制在1%以内。

2 星脉网络设计概述

2.1 系统架构

星脉网络系统架构如图3所示，它由两个关键模块组

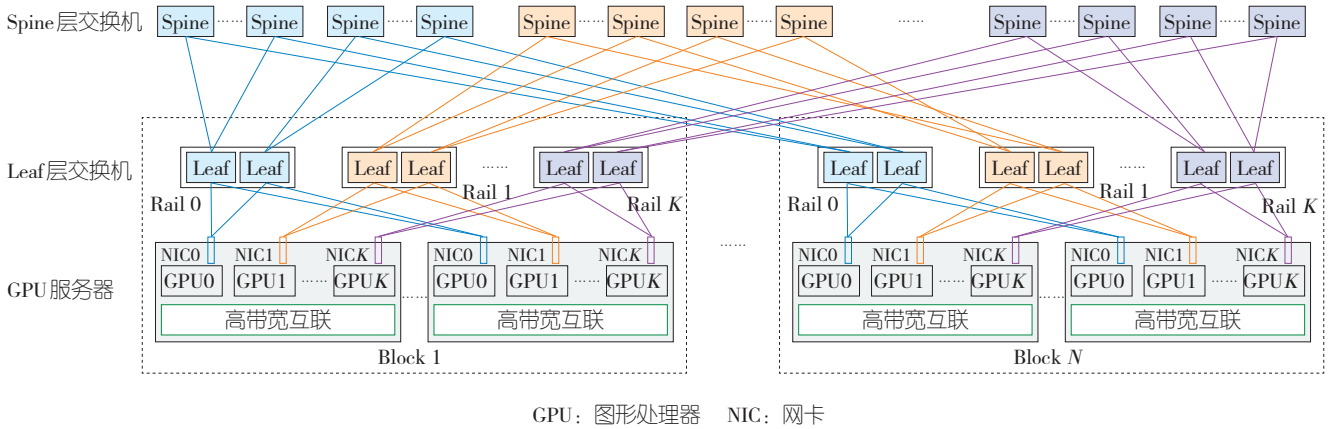


图1　GPU集群的多轨道网络架构

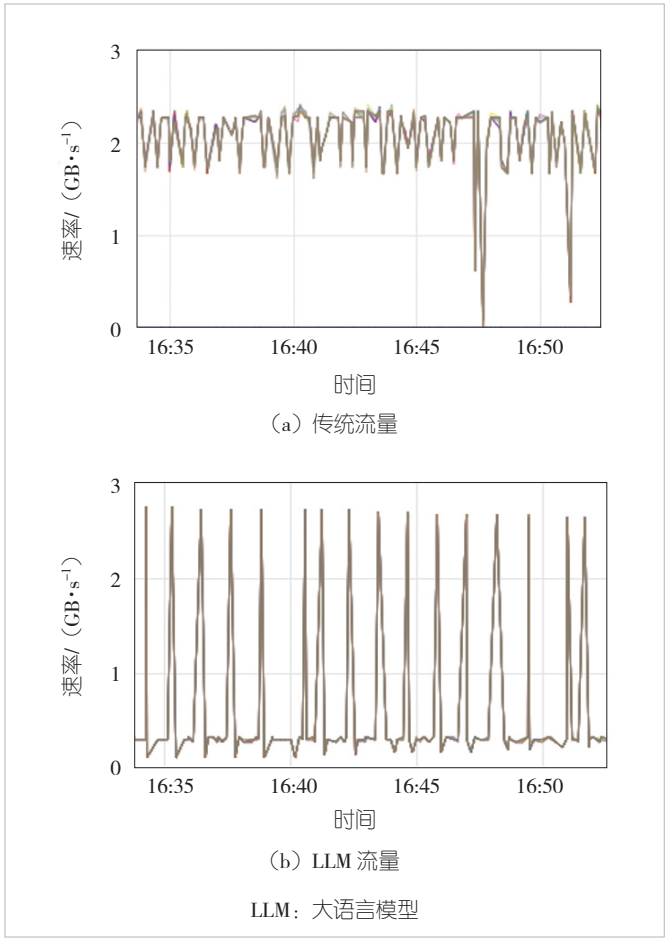


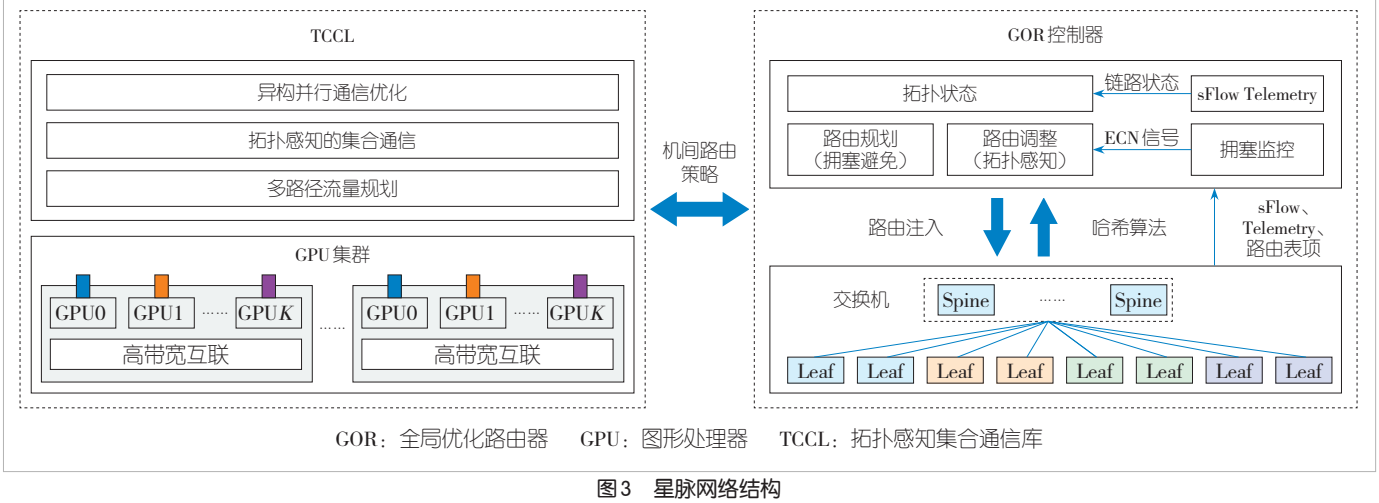
图2 从网卡出口观察到的传统流量与LLM流量

成：运行在GPU服务器上的拓扑感知集合通信库（TCCL）和管理RoCE网络中流量路由的全局优化路由器（GOR）。TCCL在GOR控制器的帮助下实现服务器内和服务器间的整体流量预规划，GOR控制器则提供拓扑信息并指导RoCE网络中的流量负载均衡。此外，GOR还负责在实际运营网络中的网络流量动态调度：首先通过网络流量和显式拥塞控制信号（ECN）来定位网络中拥塞链路，然后通过监控定位网络中的拥塞，最后通过交换机哈希注入的方式调整流量分配策略，以解决流量拥塞问题。

2.2 拓扑感知的集合通信库

TCCL针对大规模GPU集群通信场景，在标准英伟达集合通信库（NCCL）基础上实现3项核心优化：基于多级拓扑感知的通信路径优化算法、NVLink与RoCE网络的混合传输调度机制、确定性路由策略驱动的无冲突数据传输。与传统方案相比，本设计通过引入机内和机间拓扑数据感知模块，可有效弥补现有集合通信库在GPU集群网络架构感知能力上的缺陷。

当前主流集合通信库（如NCCL）依赖默认的输入服务器顺序决定集合通信路径规划。其构建典型通信路径的方式，遵循先机内通信后机间通信的原则。例如，服务器内的传输路径为：NIC0→GPU0→GPU1→……→GPU7→NIC7。跨服务器通信路径则是将每个服务器内的传输路径首尾互联，即Server1（NIC0→GPU0→……→GPU7→NIC7）→Server2（NIC0→GPU0→……→NIC7）→Server3（NIC0→GPU0→……→NIC7）。这种设计导致两个关键问题：一是相邻服务器间的NIC7→NIC0会产生跨轨道流量，这会增加网络拥塞概率；二是在多Block拓扑场景下（如Server1/3同属Block A，Server2位于Block B），默认跨机连接顺序会导致频繁的跨Block通信流量，造成二层网络中流量拥塞概率提升50%。为此，TCCL引入拓扑感知模块，通过解析服务器互联拓扑以及Block划分等特征，动态构建符合最小跳数原则



GOR：全局优化路由器 GPU：图形处理器 TCCL：拓扑感知集合通信库

图3 星脉网络结构

和轨道亲和性的通信路径。

此外，针对异构网络资源的协同利用问题，TCCL还引入了混合通信传输机制。由于机内互联网络NVLink（通信带宽为400 GB/s）和机间互联网络RoCE（400 Gbit/s × 8 NIC =3.2 Tbit/s）的通信带宽与通信协议有较大差异，TCCL引入了一种动态滑动窗口机制。该机制根据不同网络上的通信吞吐能力决定每个网络上承载的数据量，以最大化异构网络的带宽利用率。

为解决RoCE网络的哈希冲突问题，TCCL实现了确定性队列对（QP）连接管理机制。相较于NCCL随机生成QP连接参数的策略，本方案通过GOR控制器实时采集网络设备的等价多路径路由（ECMP）的哈希策略（包括源/目的IP、端口号等哈希因子），据此构建全局流表项冲突预测模型。为每个QP连接动态分配哈希正交的源端口号，可确保任意两条并发流量的哈希路径不会发生哈希冲突。

2.3 全局优化路由器

网络动态抖动会影响负载均衡，主要体现在以下4个方面：1）预规划通常根据哈希规则将流的数量均匀分布在网络链路上，但对于不同大小的流，局部拥塞仍可能发生。2）在大规模集群中，网络链路、交换机端口和GPU的故障是不可避免的，且出现故障的概率会随着设备数量的增加而增加。设备故障会导致拓扑不对称，减少可用路径，并使ECMP策略下的下一跳路径发生变化。这种路径重新分配会引发流量冲突。3）在多个模型同时运行的场景中，网络流量分布会随任务的启停时间动态变化。由于静态流量规划无法适应这种实时波动，因此在动态场景下容易出现资源竞争和流量冲突问题。4）在生产环境中，部分用户使用自定义的集合通信库（如私有消息传递接口实现），由于此类库并未纳入网络流量规划流程，可能导致协议协商失败，进而引发流量冲突。

为此，我们构建了一个基于集中控制器的动态流量管理系统。集中控制器负责实时监控网络状态，及时检测故障点并识别导致拥塞的流。控制器使用sFlow和遥测工具主动收集网络状态，并在模拟器中运用交换机厂商的实际ECMP哈希算法，以便为路由调度提供指引。当接收到拥塞信号时，全局控制器将受影响的流以最优方式调度到负载最低的路径上。

3 静态流量路由规划

3.1 LLM训练的流量特征

3D并行技术涵盖张量并行（TP）、流水线并行（PP）和数据并行（DP）技术，在LLM大模型的训练中得到广泛应用[3]。DP借助在不同GPU组上部署多个模型参数副本的方式，将数据集划分为小批量，分发给不同GPU组进行并行训练。PP和TP将单个GPU无法容纳的大模型切分到多个设备上。

在整个流程中，每个阶段需执行两次TP AllReduce操作，分别用于Attention层和多层感知机（MLP）层的前向与反向传播，使得通信频率显著高于其他集合通信类型。此外，TP AllReduce的后续计算必须依赖其通信结果才能进行，因此无法通过通信计算重叠来消除其影响。鉴于此，后续优化技术将聚焦于TP AllReduce的优化效果。

3.2 拓扑感知的集合通信调度

集合通信算法的目标是完成GPU之间的数据传输，并最大化服务器内和服务器间带宽的利用率。现有集合通信库引入了一种基于环形通信路径的AllReduce算法[4]，该算法可以充分利用服务器内和服务器间的带宽。具体过程包括3个步骤：

步骤1：构建一个服务器内的图$G(N,E)$。其中，节点集N包含服务器内的所有GPU和NIC，链路集E包括GPU–GPU和GPU–NIC链路，每条链路$e_{ij} \in E$均有一个数值表示节点i和j之间的可用带宽。例如，在图4中，NIC0与GPU0互联，NIC0和GPU0之间的可用带宽为$\min(bw_{\text{PCIe}},bw_{\text{NIC}})$，其中$bw_{\text{PCIe}}$和$bw_{\text{NIC}}$是外围组件快速互连（PCIe）链路和NIC的带宽。此外，由于所有GPU都通过NVSwitch互联，因此每对GPU之间的双向带宽都是NVLink带宽。

步骤2：基于$G(N,E)$搜索连接所有GPU的通信路径，并确定用于连接其他服务器的NIC。每条通信路径均可以表示为一个链式列表p^i_{intra}，其中$i \in [1,I]$表示第i个服务器。每条路径p^i_{intra}的头和尾用于连接其他服务器的NIC，可以表示为$p^i_{\text{intra}} \rightarrow head$和$p^i_{\text{intra}} \rightarrow tail$。每条路径$p^i_{\text{intra}}$的其他元素表示服务器内的所有GPU。例如，图4中Server1的可用路径是NIC0→GPU0→GPU1→GPU2→GPU3→NIC3。

步骤3：根据指定的服务器顺序连接在步骤2中获得的每条路径。图4中的橙色箭头表示两个服务器之间的完整路径，即 NIC0→GPU0→……→GPU3→NIC3→NIC0→GPU0→……→GPU3→NIC3。需要说明的是，图4中的通信路径仅占Server1中NIC0和NIC3的一个方向带宽。

基于环形通信路径的AllReduce算法已被证明是最优的，适用于LLM训练中的大带宽集合通信[4–5]。为了确保基于环算法的性能，服务器之间的通信需要限制在同一轨道内，以避免较长的通信长度和不均匀的负载。具体来说，跨轨道流量需要通过骨干交换机和其他轨道上的Leaf交换机。此外，跨轨道流量可能与现有流量发生冲突，导致两者速度减慢，

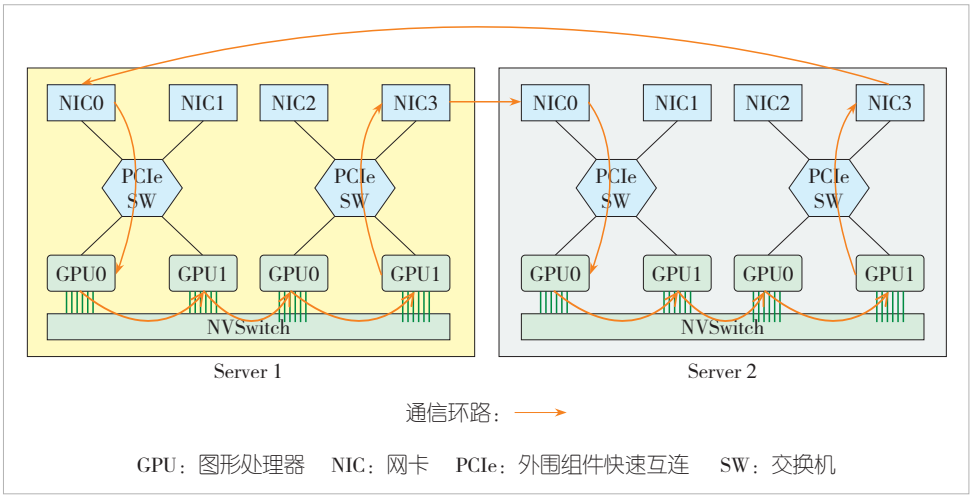


图4 服务器间构建通信环路

从而导致集合通信性能下降。然而，NCCL在通信路径规划的具体实现中并没有考虑轨道网络架构的影响。例如，NCCL按照服务器顺序构建了一个通信路径：Server1（NIC0–GPU0–GPU7–NIC7）→Server2（NIC0–GPU0–GPU7–NIC7）→Server3（NIC0–GPU0–GPU7–NIC7）→Server4（NIC0–GPU0–GPU7–NIC7）。其中，节点间连接都是通过跨轨道的NIC0（轨道0）和NIC7（轨道7）互联实现的，这会导致所有跨节点流量均为跨轨道流量。

由于交换机的端口数量有限（在我们的测试场景中，一个交换机有128个端口），每个Block只能容纳32个配备有8个NIC的服务器。随着集群规模的增加，Block的数量不可避免地会增加。Block之间通过Spine交换机实现互联，因此跨Block的流量必须通过Spine交换机。跨Block流量增多会导致服务器之间的通信时延增加、Spine层的交换机拥塞概率增大。因此，在决定服务器之间的通信顺序时，需要考虑服务器在Block中的分布。然而，NCCL无法感知服务器在网络架构中的分布情况，因此通常默认根据给定的服务器列表顺序确定服务器通信顺序。然而，这会额外产生大量不必要的跨Block流量。例如，如果遵循给定的服务器顺序，Server1（Block1）→Server3（Block2）→Server2（Block1）→Server4（Block2）→Server1（Block1），跨Block流量的比例为100%。然而，如果同一Block中的服务器完成顺序连接，跨Block流量的比例将下降到50%。

为了解决上述集合通信算法的问题，本文提出了一种分层拓扑感知通信路径规划算法，以优化基于环形通信路径的AllReduce算法，使其适应多轨道网络架构并减少跨轨道流量。具体来说，首先确保所有服务器的最佳通信顺序，然后使用同一轨道中的NIC来连接服务器内的通信环，以避免跨轨道流量。基于此，我们首先找到所有服务器所在的Block，并将属于第k个Block的所有服务器聚类到集合$block_k$中；然后，对集合$\{block_k, k \in \{1,\cdots,K\}\}$中的所有服务器进行排序，以获得最佳服务器通信序列，其中K是集群中的块数；之后，遍历所有轨道$\{rail_j, j=\{1,\cdots,J\}\}$，在每个服务器上找到属于同一轨道$rail_j$的NIC，并将其作为服务器内通信路径的头和尾，以避免跨轨道流量；最后，将所有服务器内通信路径$\{p^i_{intra}, i \in 1,\cdots,|\Psi|\}$与$p^i_{intra} - 1 \rightarrow next = p^i_{intra} \rightarrow head$连接，形成一个完整的通信环，其中$|\Psi|$是最佳服务器通信序列$\Psi$的长度。

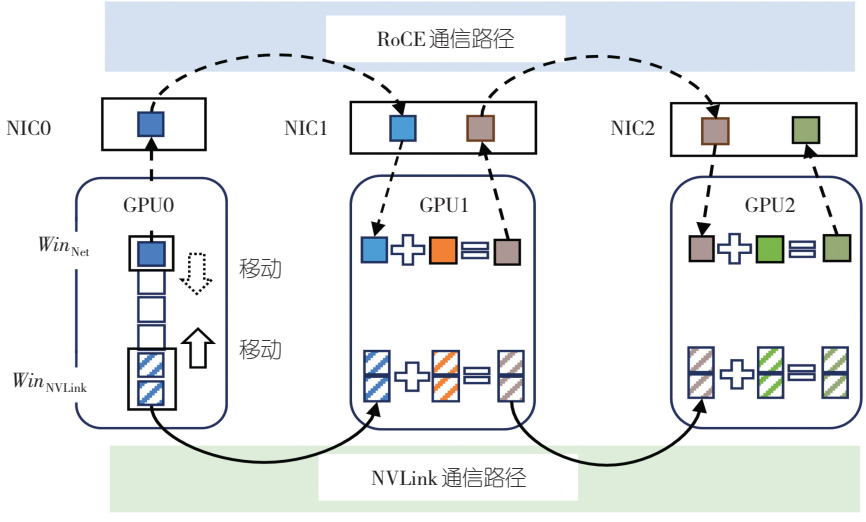
3.3 RoCE和NVLink并行通信优化

LLM模型通过TP和PP拆分为多个部分，并部署在不同的GPU上。每个TP组和PP组利用集合通信（例如AllReduce）在模型层内和层间实现参数传输。由于每个模型层在前向和后向过程中需要4次TP AllReduce，根据GPT–3175B的并行设置，TP AllReduce的总次数可以达到6 720次，其中每次TP AllReduce的通信量可以达到224 MB。为了减少通信延迟，现有工作将每个TP组限制在服务器内，以充分利用高速互联（例如NVLink）。然而，如图3所示，我们的GPU服务器不仅配备了高速互联，还为服务器中的每个GPU配备了一个NIC。这意味着服务器内的TP AllReduce可以通过NVLink以及绑定到GPU的NIC完成。对此，我们提出了一种异构通信优化方案，该方案可利用空闲RoCE网络来增强服务器内的集合通信（例如TP AllReduce），进一步提高系统训练性能。

为了实现服务器内和服务器间的多路径并行通信，常规做法是将发送的消息拆分为多个块，然后轮询每条路径发送块，并在路径末端组合块以获得最终结果。然而，TP All-Reduce需要在每个GPU上同步一次传输的结果以进行加法操作。整个集合通信的延迟取决于最慢的路径。需要注意的是，NVLink和RoCE网络的传输延迟分别为200 ns和4 μs。这导致通信性能受限于RoCE网络的延迟。这一问题的根本缘由在于，两条通信路径没有完全解耦，需要在每次传输后完成同步操作。因此，我们提出在集合通信粒度上切片TP

AllReduce，将每个GPU上要传输的消息分为两个部分，分别用于NVLink通道和网络通道。每部分消息可以独立通过NVLink或网络通道传输，以获得最终的集合通信结果。图5展示了在两个通道上并行传输数据的操作原理。GPU0中的蓝色方块数据可以通过NIC0和NIC1之间的RoCE网络传输到GPU1，并与GPU1中的相应橙色方块完成加法操作。GPU2将重复相同的操作以获得最终AllReduce结果。带有蓝色条纹的方块也可以实现NVLink通道上的相同AllReduce过程。这两个通信过程完全独立，以确保NVLink和网络通道都能实现线速传输。

此外，还有一个重要的问题是可以将多少数据卸载到网络通道上，以便提高服务器内集合通信的性能。因此，卸载到网络通道的数据，需要在NVLink通道完成剩余数据传输任务之前完成传输。显然，基于网络通道和NVLink通道的静态延迟比值来确定卸载到网络通道的数据量是不可能的。这是因为背景流量和异步通信等因素导致两个通道的状态动态变化。为此，我们设计了一种动态滑动窗口机制，以便动态调整在两个通道上传输的数据量。具体来说，我们首先将发送的消息分为m个块，并根据NVLink通道和网络通道的带宽计算传输窗口的大小（即Win_{NVLink}和Win_{Net}），以分配每个通道上的数据。例如，如果NVLink通道和网络通道的AllReduce带宽分别为100 GB/s和50 GB/s，那么Win_{NVLink}和Win_{Net}可分别设置为2个块和1个块。如图5所示，NVLink通道和网络通道的初始传输窗口分别被设置为待传输数据的头和尾。之后，两个通道开始并行传输相应窗口中的数据。当任一通道的窗口数据传输完毕时，窗口滑动Win_{NVLink}或Win_{Net}以准备下一次传输。持续重复这一过程，直到两个通道传输的块数累计达到m个块。



GPU：图形处理器　NIC：网卡　RDMA：远端直接内存访问　RoCE：基于融合以太网的RDMA

图5　RoCE网络和NVLink网络并行通信优化

3.4 网络流量规划

网络流量规划的目标是，通过均衡交换机间的负载分配避免潜在的瓶颈或拥塞。为实现高可用性，NIC采用了双端口上联两个Leaf交换机的模式。为了充分利用NIC和两个Leaf交换机之间的带宽，我们计划在每个源地址与目的地址间建立至少两条流（例如设计2个QP，实际使用4个QP）。此外，交换机通常采用ECMP的方式进行流级负载均衡，并根据五元组哈希值将流量分配到不同路径。实际上，大多数拥塞问题源于多个流被哈希到同一路径上。

1） 路由注入策略的选择

一种可能的方法是通过向交换机注入路由策略来规划流路径。基于策略的路由（PBR）可通过定义五元组流的下一跳建立访问控制表（ACL）策略，并通过全局规划最小化流路径冲突。然而，该方法存在两个缺陷：一是，交换机的ACL规则数量通常有限，而全局规划需在每个交换机部署大量的ACL规则；二是，管理过多ACL规则对网络维护人员而言是巨大负担。因此，我们通过特定流源端口来规划路径以最小化冲突。

2） 源端口号的确定

若未理解ECMP路由算法，通信库只能借助随机端口号使流随机选择路径。这种方法无法保证多流均匀分布，也无法充分利用大规模网络中的可用路径。实验表明，此类不均衡可能导致可用带宽下降25%。现有工作[6]证明，基于ECMP哈希函数的线性特性可避免同设备对之间多个QP的路径冲突。假设ECMP哈希函数为$H(p)$（p为流源端口），其线性特性表现为$H(p\oplus\delta)=H(p)\oplus H(\delta)\oplus H(0)$。因此，对于源端口$p$和$p\oplus\delta$，存在如下等式关系：$H(port)\oplus H(port\oplus\delta)=H(\delta)\oplus H(0)$。由此可知，$H(p)$与$H(p\oplus\delta)$是否相等仅由$\delta$决定。为确保$H(p)\neq H(p\oplus\delta)$（即两流使用不同路径），只需要$H(\delta)\neq H(0)$即可。

在实际操作中，可通过各交换机的哈希算法模拟器，固定源端口号（如50000）并顺序调整目标端口号，观察哈希值变化。通过遍历目标端口号范围（如1～65536），筛选出4个编译量δ（如δ = 1~4），使得$H(\delta) \neq H(0)$。这4个目的端口号将分配给4个QP（即4条流），以避免流量冲突。

4 动态网络流量调度

尽管路由规划借助源端口多路径分发机制确保了流量的均匀分布，但在实际应用场景中仍存在一些问题：1）考虑到高性能网络的规模，链路中断导致网络拓扑结构改变的情况普遍存在；2）预规划旨在通过哈希规则将流数量均匀分配到各网络链路，但由于流大小存在差异，当大流量被分配到同一条链路时仍可能会引发拥塞；3）在需要多任务并行部署的云环境中，新任务的到达具有不可预知性。因此，仅依赖流量预规划无法有效解决上述问题。

星脉网络能够通过集中式控制器来应对动态流量调度中的双重挑战：

• 实时拥塞检测：基于分布式探针和流量特征分析，识别毫秒级拥塞事件；

• 全局动态路由规划：利用在线优化算法动态调整流量分配策略。

星脉网络的核心创新在于将调度周期压缩到一个LLM训练迭代周期（典型值为10～30 s）内，在当前迭代内完成下一个迭代的流量调度和资源预留，从而抑制流量拥塞跨训练迭代传递。

4.1 网络拥塞监测

基于对LLM训练过程日志的分析，我们发现网络事件可以分为两类：78%表现为连接中断，22%表现为因链路拥塞导致的性能下降。实时监控网络状态对于及时检测拥塞和定位受影响数据流具有关键作用。

在高性能网络场景中，LLM训练流量呈现高突发性，链路拥塞持续时长通常在数十毫秒至数百毫秒量级。实验证实，秒级粒度的链路状态观测无法有效捕捉实际发生的拥塞事件（这些拥塞已显著影响训练收敛性）。然而，采用更细粒度的统计窗口（如亚秒级采样）会引发两个技术难题：一是，系统日志量存呈指数级增长；二是，现有硬件设备不支持高频数据采集。因此，本文提出基于ECN的拥塞感知方法，通过优化拥塞指标敏感性解决上述矛盾。

ECN是数据中心传输控制协议、数据中心量化拥塞通知协议等拥塞控制协议的核心机制[7]。在实际部署中，系统以10 s为周期统计交换机端口的ECN标记数量，并定义三级告警规则：若1 min内ECN标记数连续3次超过500阈值，则触发中级告警；若相邻LLM训练阶段间隔（典型值为10～30 s）内累计告警次数≥5次，则触发高级告警。该设计在保证采样频率与LLM训练流量周期对齐的同时，减小了误告警出现的概率。

4.2 动态场景下全局路由规划

在检测到由流量冲突引起的中、高级ECN告警后，我们通过sFlow流量采样技术[9]和Telemetry网络遥测技术[10]识别待重路由的流量，并通过哈希注入和路径模拟来选择最优路径。具体流程如下：

1）流量识别：当交换机触发ECN告警时，sFlow从告警端口采集所有流量的五元组信息，并以10 s为一个周期持续执行6次连续状态查询。按平均带宽降序排序后，选取在6个周期中出现≥2次的Top N流量，记为集合F。

2）流量上下文解析：对于每个流$Flow_i \in F$，提取其源GPU_i、连接的Leaf交换机$leaf_k$、在$leaf_k$上的出端口GE_{ki}，并构建集合C——包含$leaf_k$上所有与$Flow_i$具有相同目标IP但源IP不同的流（不包含$Flow_i$自身）。由于交换机上的流量路由控制粒度不是针对特定流的，而是针对具有相同目标IP的所有流量，因此，我们需要考虑集合C中所有流量在新路由策略下是否可以避免流量冲突。

3）下一跳探测：通过遥测技术查询$leaf_k$上$Flow_i$目标IP可用的下一跳（排除GE_{ki}）及其空闲带宽，并按空闲带宽降序排列形成有序列表L。

4）路径重计算：

• 哈希模拟：提取交换机的五元组哈希算法，构建路径计算模拟器。

• 路径映射：对于集合C中的每个流$Flow_k$，使用模拟器计算其在下一跳集合中命中的出口端口，并确定该端口连接的$Spine_j$交换机。

• 递归验证：基于$Flow_k$目标IP查询$Spine_j$的路由表，确定其下一跳集合H_j，随后递归式地找出$Flow_k$经过的所有物理链路路径。

• 冲突检测：在确定$Flow_k$的所有物理链路路径后，执行链路状态检查，具体包括：（1）优先从控制器内存中获取实时链路占用数据；（2）若内存无数据，则通过Telemetry系统实时查询；（3）若$Flow_k$带宽叠加后，链路的总占用率超过其容量的75%，则终止当前迭代并返回步骤。

5）路由更新：将验证通过的路由策略同步交换机控制面，完成ECMP表项更新。

实验数据表明，该机制可显著缩短网络拥塞持续时间。在实际测试中，拥塞时长降幅超过80%（$p<0.01$），满足AI训练场景对微秒级拥塞响应的需求。

5 星脉网络性能评估

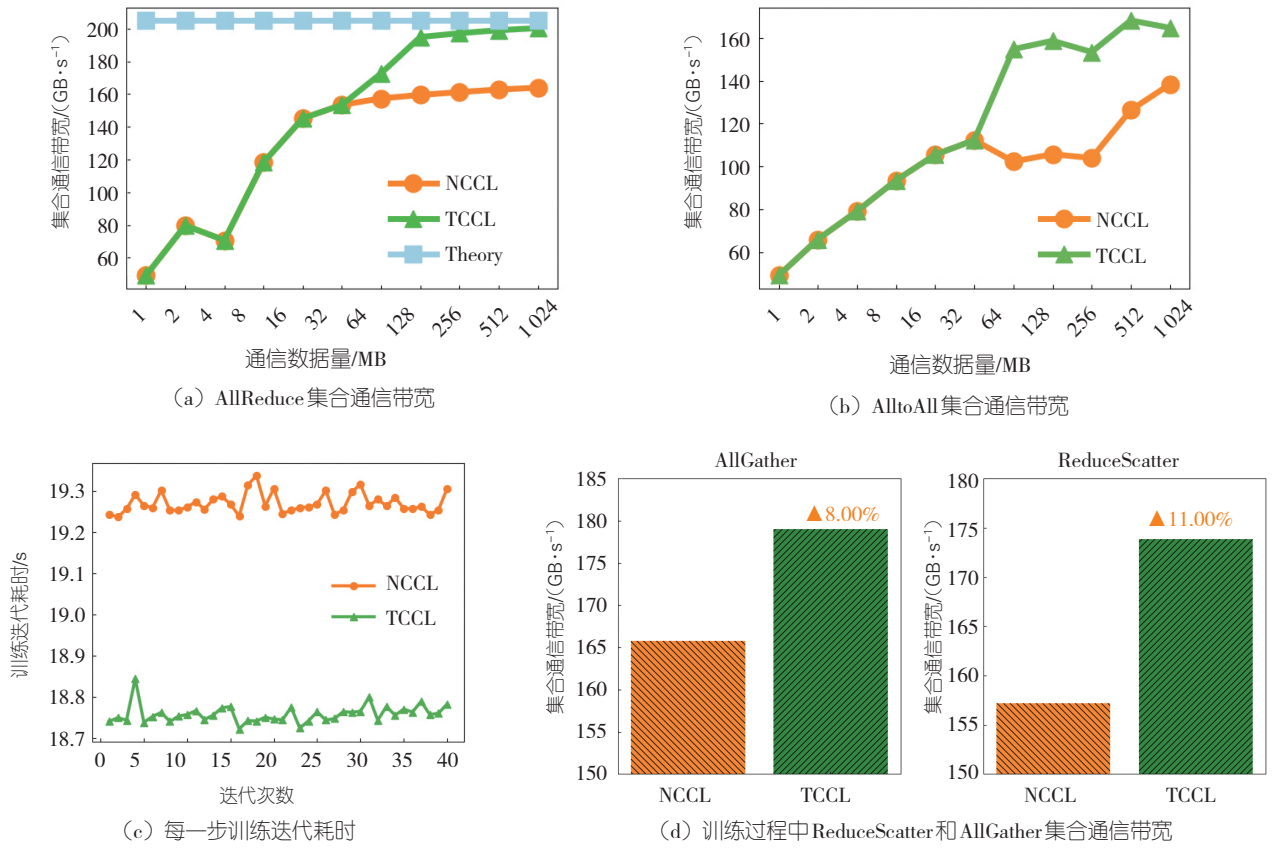
5.1 评估环境设置

我们通过一个NVIDIA GPU集群对星脉网络进行性能评估。该集群的网络采用8轨道网络架构设计。每块GPU的NVLink高速互联总线通过NVSwitch交换芯片与机内其他7个GPU互联，并同时与一个双端口Mellanox ConnectX-6/7智能网卡绑定。跨机网络基于RoCEv2协议构建，通过PFC流控机制实现无损传输。负载均衡采用GOR策略，同时引入DCQCN，并结合动态ECN配置[11]，实现网络拥塞控制的管理。

5.2 TCCL的性能评估

我们采用标准NCCL-TEST测试工具[5]对TCCL和开源NCCL（评估中采用2.17.1版本）进行性能评估。实验采用NCCL-TEST定义的全规约总线带宽（简称Busbw）作为核心指标，其计算公式为：$\text{Busbw} = \frac{2(n-1)}{n} \times \text{Algbw}$，其中，$n$表示GPU的数量，Algbw为算法带宽（可通过通信信息量与集合通信耗时）。总线带宽Busbw能更准确地反映集合通信对物理带宽的实际利用率，因此，后续实验结果均采用该指标进行量化分析。

图6（a）中蓝色线的性能上限表示，在1 Gbit/s通信量下AllReduce的集合通信带宽的理论上限（单机8个网卡带宽之和，即200 Gbit/s）。在相同条件下，TCCL几乎达到该理论极限，较NCCL提升了22%。对于64 MB及以上的通信量，采用异构通信优化的TCCL始终优于NCCL，性能提升幅度为9%～25%。然而，当通信量小于64 MB时，TCCL与NCCL性能相近。这是因为此时性能受限于通信网络时延，所有的数据都会优先通过NVLink网络传输（NVLink的传输时延为200 ns，远小于RoCE的传输时延2 μs）。这意味着TCCL方案效果和NCCL方案一致，因此两者最终的AllReduce Busbw性能也一致。异构并行策略优化不仅对AllRe-



（a）AllReduce集合通信带宽　（b）AlltoAll集合通信带宽

（c）每一步训练迭代耗时　（d）训练过程中ReduceScatter和AllGather集合通信带宽

NCCL：英伟达集合通信库　TCCL：拓扑感知集合通信库

图6　TCCL异构并行通信优化的性能评估

duce场景有效，还能提升其他集合通信操作（如AlltoAll）的性能。如图6（b）所示，在1 GB通信量下采用异构并行优化的AlltoAll性能提升了20%。

为进一步验证该优化的有效性，我们在64台H800服务器上开展对比实验，分别使用TCCL与NCCL对类GPT模型进行训练，并对两者的耗时情况进行对比分析。如图6（c）所示，采用异构通信优化的TCCL在每轮训练迭代中平均节省约2.5%的时间。此外，运行时对TP AllReduce的带宽监测显示，AllGather和ReduceScatter操作分别实现了8%和11%的性能提升。这些结果充分证明了异构通信优化在不同类型集合通信操作中的广泛有效性。

图7展示了拓扑感知路由在不同节点规模下的性能增益（测试通信量大小为1 GB）。由图7（a）可知，随着节点数量增加，拓扑感知路由带来的性能提升显著增长：当节点规模达到140时，性能提升幅度可达13%。这是由于节点规模扩大导致通信节点数量增加，NCCL方案会产生跨Spine交换机流量，加剧流量负载不均衡。因此，在大规模集群中，需通过精细的流量规划来避免非必要的跨Spine交换机通信。

随着节点数量的增加，拓扑感知路由的性能增益也在增加。当节点数量达到140时，系统性能将提升13%。这是因为，随着节点数量的增加，涉及的块数量也随之增加。NCCL引入了跨Spine流量，导致流量负载更加不平衡。因此，在大规模集群中，需要仔细规划机器之间的流量，以避免不必要的跨Spine流量。

为了评估分层拓扑亲和流量规划对集体通信稳定性的影响，我们从100台机器中随机选择4台进行200次重复实验，对比TCCL和NCCL的AllReduce Busbw性能（见图7（b））。统计分析表明，TCCL与NCCL的平均集合通信性能均稳定在11.5 GB/s。NCCL在200次实验中的标准差为1.26，显著高于TCCL的0.31。该差异源于NCCL存在大量的跨Spine流量，这种情况增加了负载不均衡概率，从而导致通信性能的波动。

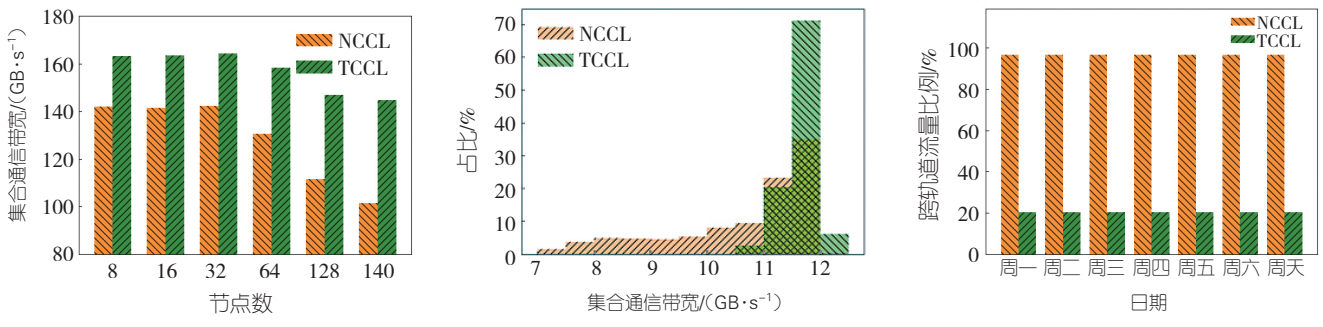
在GPT3模型训练任务中部署TCCL后，我们持续监测一周的跨Spine流量（结果见图7（c））。相比于未采用TCCL的方案，该方案的跨Spine流量减少了75%。流量优化有效保障了网络负载均衡，同时使GPT3的样本训练吞吐量从NCCL方案的45个/s提升至49个/s。

5.3 GOR的性能评估

我们首先评估了GOR控制器在64 MB～1 GB通信数据量下对AllReduce任务的动态流量调度能力。实验表明，无论通信量多大，都会触发流量拥塞告警。这是因为，集合通信的流数量由GPU数量和QP队列对数决定。同时我们发现，大部分拥塞告警是由哈希冲突引发的。为此，我们启用GOR对所有冲突流量进行调度，直至消除所有拥塞告警。图8（a）展示了调度前后的测试结果，可以看出AllReduce的Busbw性能提升了8.4%～18.8%。

我们随后在AlltoAll场景中评估了GOR控制器的性能。由于AlltoAll涉及所有参与网卡间的通信并产生大量数据流，其拥塞概率会显著增加，同时流量调度也更为复杂。图8（b）显示，当通信量为64 MB且关闭GOR时，AlltoAll的Busbw达到最低值。这是因为该配置触发了DCQCN水位线机制并产生大量ECN，导致网卡降速。在启用GOR后，控制器将拥塞流量调度至相对空闲的路径，使Busbw性能提升77.4%，如图8（b）所示。

在运营网络验证中，图8（c）显示启用GOR调度后，集群拥塞告警数量与持续时间均下降80%。系统稳定后，每



（a）AllReduce集合通信带宽　（b）200次试验集合通信带宽分布　（c）每日跨轨流量占比统计

NCCL：英伟达集合通信库　TCCL：拓扑感知集合通信库

图7　TCCL拓扑感知流量路径规划的性能评估

日告警量维持在10次左右，每次可通过1～2次调度在10 s内解决。告警数量的波动源于新任务到达时的流量模式变化，而持续观测结果验证了动态流量调度的稳定性。

图9（a）展示了单端口场景下GOR的拥塞消除效果。调度后交换机出口的ECN计数器归零，表明拥塞即时解除。图9（b）则呈现了多并发流导致的严重拥塞处理过程。通过三轮调度（将最大流量依次迁移至Link1和Link2），ECN通知量从初始的10 000以上逐步降至500以下，最终有效消除了链路拥塞。

6 结论

本文介绍了星脉网络，一个用于万卡GPU集群互联的高性能无损网络实践案例。星脉网络采用端侧集合通信库与全局路由控制联合优化方案，使AllReduce和AlltoAll通信带宽性能相较于公开最优方案（NCCL）分别有25%和22%的提升。网络测试结果表明，星脉网络能够有效管理LLM大模型训练中的大带宽和高突发的流量，把网络拥塞概率降低到1%以内，使网络拥塞时长下降80%。

参考文献

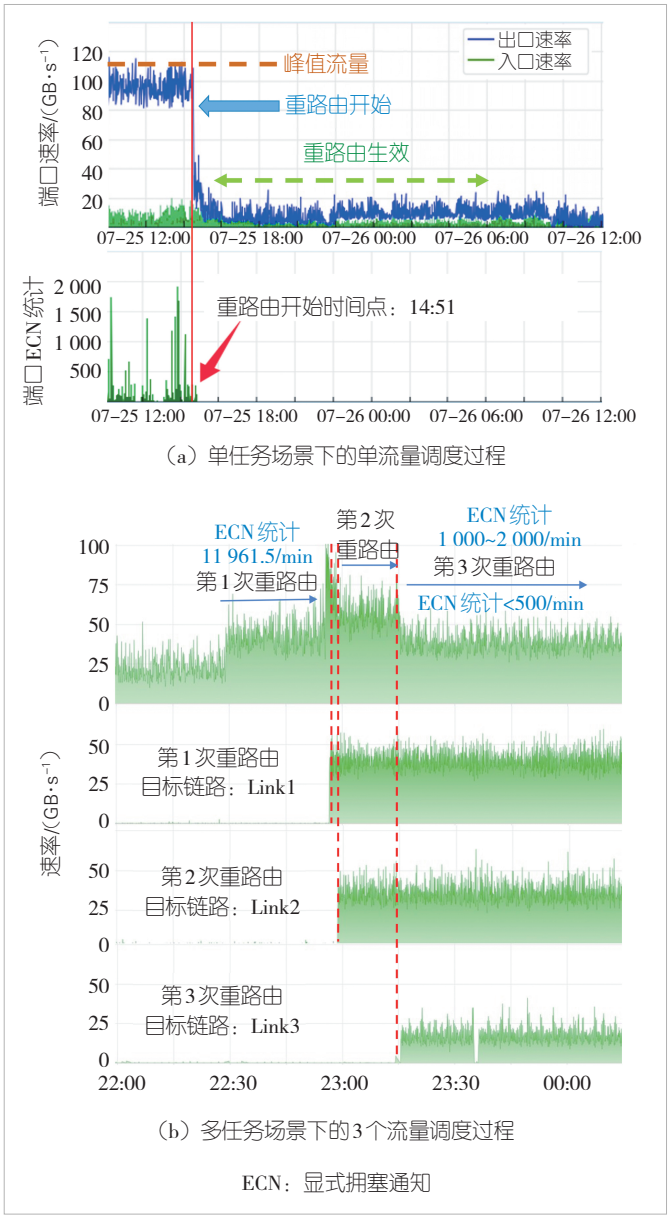
[1] YENDURI G, RAMALINGAM M, CHEMMALAR S G, et al. Generative pre-trained transformer: a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions [EB/OL]. [2025-03-10]. https://arxiv.org/abs/2305.10435v2

[2] GAO Y, LI Q, TANG L, et al. When cloud storage meets RDMA [EB/OL]. [2025-03-10]. https://www. usenix. org/system/files/nsdi21-gao.pdf

[3] SONG J, YIM J, JUNG J, et al. Optimus-CC: efficient large NLP model training with 3D parallelism aware communication compression [EB/OL]. [2025-03-10]. https://arxiv. org/abs/2301.09830v1

[4] NVIDIA Developer. NVIDIA collective communications library (NCCL) [EB/OL]. [2025-03-10]. https://developer.nvidia.com/nccl

[5] Github. NCCL tests [EB/OL]. [2025-03-10]. https://github. com/NVIDIA/nccl-tests/tree/master

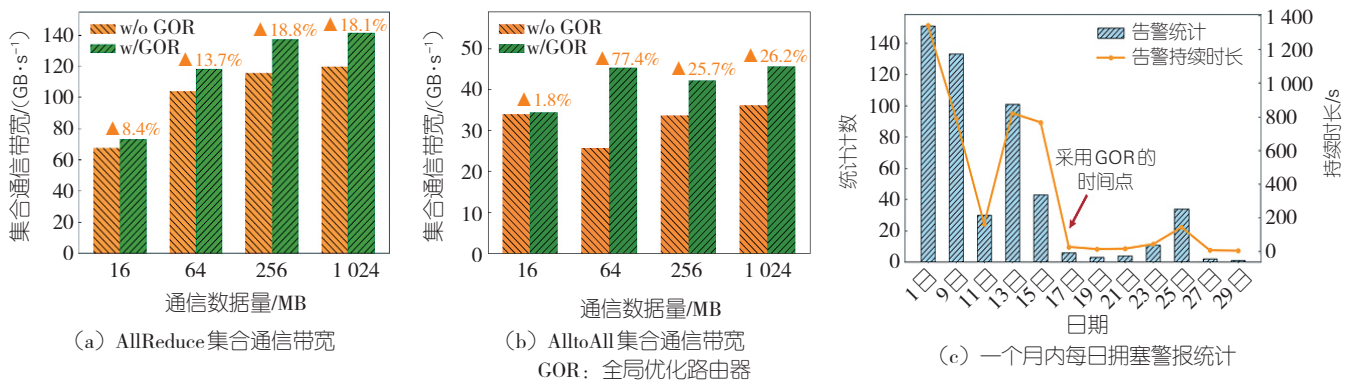


（a）单任务场景下的单流量调度过程

（b）多任务场景下的3个流量调度过程

ECN：显式拥塞通知

图9 全局优化路由器（GOR）消除流量拥塞过程



（a）AllReduce集合通信带宽

（b）AlltoAll集合通信带宽

（c）一个月内每日拥塞警报统计

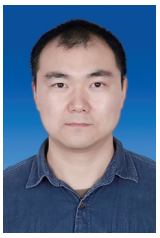
GOR：全局优化路由器

图8 使用GOR控制器的动态流量调度性能评估

[6] ZHANG Z, ZHENG H, HU J, et al. Hashing linearity enables relative path control in data centers [EB/OL]. [2025-03-10]. https://www.usenix.org/conference/atc21/presentation/zhang-zhehui
[7] KUMAR G, DUKKIPATI N, JANG K, et al. Swift: delay is simple and effective for congestion control in the datacenter [C]// Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. ACM, 2020: 514-528. DOI: 10.1145/3387514.3406591
[8] NARAYANAN D, SHOEYBI M, CASPER J, et al. Efficient large-scale language model training on GPU clusters using megatron-LM [C]//Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. ACM, 2021: 1-15. DOI: 10.1145/3458817.3476209
[9] WANG M, LI B, LI Z. sFlow: towards resource-efficient and agile service federation in service overlay networks [C]//Proceedings of 24th International Conference on Distributed Computing Systems, IEEE, 2004: 628-635. DOI: 10.1109/ICDCS.2004.1281630
[10] YU M L. Network telemetry [J]. ACM SIGCOMM computer communication review, 2019, 49(1): 11-17. DOI: 10.1145/3314212.3314215
[11] YAN S Y, WANG X L, ZHENG X L, et al. ACC: automatic ECN tuning for high-speed datacenter networks [C]//Proceedings of the 2021 ACM SIGCOMM 2021 Conference. ACM, 2021: 384-397. DOI: 10.1145/3452296.3472927

作 者 简 介

李宝嘉，腾讯数据中心网络架构师；主要负责AI大模型超算网络架构方案论证和设计、训练架构加速方案优化以及集合通信优化。



何春志，腾讯数据中心网络架构师；主要负责腾讯高性能计算网络的架构设计、集合通信库/网络协议研究，以及大模型训练推理业务与高性能网络的联合协同优化。

夏寅贲，腾讯网络首席架构师；主导腾讯星脉网络系统的设计研发工作，构建从自研软硬件系统到端到端AI集群运维的高性能网络系统，支撑腾讯多个万卡AI集群的快速建设与高效运行。

何泽坤，腾讯基础网络中心总监，并担任开放数据中心委员会（ODCC）网络工作组组长；长期深耕数据中心网络、骨干网络架构，主导腾讯全球数据中心互联网络及与运营商互联网络的设计，近年来聚焦AI算力网络协同创新，牵头开放数据中心委员会ETH-X开放超节点与MegaScaleOut项目。

王晓亮，南京大学计算机学院副教授；长期从事网络体系结构的研究工作；曾获APNET、BIGCOM、EuroSys最佳论文奖，并获得2019年江苏省科技进步奖一等奖、2023年江苏省科技进步奖二等奖；发表论文50篇。