多芯粒大模型加速器推理协同优化方法

方娟，潘晨阳，古明辉，李硕朋，陈慧杰，翟冉
（北京工业大学计算机学院，北京 100124）

摘　要：在采用2.5D封装集成多计算芯粒与存储芯粒的大模型推理加速系统中，模型推理解码阶段跨芯粒通信具有突发性与强非均衡性，流量在拓扑中聚集到少数链路并形成热点排队，封装内网络通信常成为性能瓶颈。为缓解上述瓶颈，提出T²-CHIP协同优化方法，通过刻画解码阶段跨芯粒通信在互连中的分布特征，识别热点链路，对带宽资源重分配，同时调整任务映射以减少热点跨芯粒交互，从而有效缓解解码阶段通信拥塞。周期精确网络仿真结果表明，该方法在提升解码阶段尾部性能与整体吞吐量的同时，降低了动态功耗，且维持了较低的实现开销。

关键词：大语言模型；2.5D芯粒架构；芯粒间互连；异构协同优化
中图分类号：TN47；TP302.1
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2026042

Cooperative optimization method for inference on multi-chiplet large-model accelerators

Fang Juan, Pan Chenyang, Gu Minghui, Li Shuopeng, Chen Huijie, Zhai Ran
College of Computer Science, Beijing University of Technology, Beijing 100124, China

Abstract: In large language model inference accelerators integrating multiple compute and memory chiplets via 2.5D packaging, cross-chiplet communication during the decoding stage is bursty and highly unbalanced, causing traffic to concentrate on a small number of links and form hotspot queuing, which tends to make the network-on-package a performance bottleneck. To mitigate this bottleneck, T²-CHIP was proposed, a collaborative optimization method that characterized the distribution of decoding-stage cross-chiplet traffic over the interconnect, identified hotspot links, reallocated bandwidth resources, and adjusted task mapping to reduce hotspot cross-chiplet interactions, thereby effectively relieving decoding-stage communication congestion. Cycle-accurate network simulations show that the proposed method improves decoding-stage tail performance and overall throughput while reducing dynamic power consumption and maintaining low implementation overhead.

Keywords: large language model, 2.5D chiplet architecture, die-to-die interconnect, heterogeneous co-optimization

0　引言

近年来，大语言模型（large language model，LLM）已显示出模型规模扩展带来的能力跃迁，并在搜索问答与代码生成等场景中得到广泛应用。随着模型规模增大与上下文长度增加，推理阶段对计算、存储与互连的需求同步增长，系统性能愈发依赖资源间的协同匹配。受制于版图尺寸、布线复杂度、功耗与良率等因素，单芯片规模难以支撑推

收稿日期：2025–11–25；**修回日期：**2026–02–05
通信作者：李硕朋，lishuopeng@bjut.edu.cn
基金项目：国家自然科学基金资助项目（No.62202019, No.61202076）；北京市自然科学基金资助项目（No.4262021）
Foundation Items： The National Natural Science Foundation of China (No. 62202019, No. 61202076), Beijing Natural Science Foundation (No.4262021)

理系统扩展[1]。芯粒是一种通过先进封装技术集成不同工艺节点的芯片裸片，形成多功能模块化芯片的技术体系，为算力与存储的横向扩展提供现实路径。多芯粒大模型加速器指的是通过先进封装将多个计算芯粒与存储芯粒等集成于同一封装内，并依赖封装内网络（network on package，NoP）互连完成跨芯粒数据传输以支撑 LLM 推理的专用加速器，本文聚焦于 2.5D 封装的多芯粒加速器形态。

芯粒可以将不同工艺制程的芯片集成在一起，因而对高端工艺依赖少，有着高良率、高灵活性以及低成本等优势，但同时芯粒间跨芯粒通信受限于封装内互连带宽与能效，传输时延远高于芯粒内时延，成为通信瓶颈，尤其是在大模型推理场景中，如何实现芯粒间的高效通信成为一个重大技术难题[2-7]。大模型推理过程通常可划分为预填充与解码两个阶段，这两个阶段在并行度与资源瓶颈上存在显著差异，预填充阶段具有较高并行度，主要受计算与存储带宽约束；解码阶段逐词元（token）生成，注意力计算与键值缓存管理对数据访问与搬运高度敏感[8-10]。模型在线推理服务通常用窗口化尾部指标刻画稳定性，其中窗口化 p99（99th percentile）是常用度量之一[9]，反映的是时延分布的高分位长尾特征。在芯粒系统中，前述敏感性会进一步体现为细粒度流量在拓扑中非均衡聚集并形成热点排队，增加尾部时延并削弱服务稳定性。

围绕多芯粒推理优化，既有研究主要通过算子划分、任务映射与调度降低跨芯粒通信量，或通过功能异构化改变计算与存储分工，从而提升平均性能或总体吞吐量[3-7]。然而，这类以平均性能为主的目标存在两方面局限。第一，解码尾部行为由少数热点链路主导，平均指标改善难以保证窗口化 p99 时延的同步收敛，也难以直接约束最忙链路利用率。第二，在平台复用约束下，现有方案多作用于需求侧，对供给侧能力调整有限。芯粒的巨大优势和重要价值之一在于模块化复用，在同一封装平台上复用既定芯粒与接口资源，可在较低的一次性工程（non-recurring engineering，NRE）费用与平台迭代成本下覆盖不同模型规模与服务场景。工程上也通常更倾向于复用既定的 NoP 拓扑、路由与路由器/物理层（PHY）微结构，并将封装内的 SerDes 通道总量与总 D2D 带宽预算视为既定，从而避免代价高昂的芯粒与互连硬件重设计。因此，系统的优化空间主要在封装内互连带宽的逐链路配置以及与之匹配的映射对齐上。由此引出本文研究问题：在不改变既定拓扑、路由器微结构与总带宽预算的前提下，能否通过面向解码流量特征的逐链路带宽分配与任务角色对齐，使解码阶段窗口化尾部时延更稳定地收敛，并在满足最忙链路利用率约束的同时兼顾首 token 时延与吞吐量。

针对上述问题，本文提出 T²-CHIP——一种面向 2.5D 异构多芯粒大模型推理的协同优化方法。在固定网格拓扑与统一路由器微结构前提下，T²-CHIP 通过分析解码阶段跨芯粒通信在互连中的分布特征识别热点链路，在总带宽预算守恒约束下进行带宽重分配，并结合任务角色对齐减少热点跨芯粒交互。该问题的挑战在于解码流量具有显著的 ON/OFF 突发性与强空间非均衡性，且瓶颈截面会随模型规模、上下文长度与负载形态漂移，使仅依赖平均指标或静态经验配置难以稳定控制窗口化 p99。为应对解码流量的突发性与非均衡性，本文在尾时延代理中引入链路级等效突发因子，并采用通道粒度的闭环校正以抵消离散实现偏差；同时，针对瓶颈截面漂移，提出基于热点集合重合率的触发式更新策略；此外，通过分层画像与契约约束给出一致统计口径，结合连续优化与离散化映射获得可部署方案，实现严格预算下的拥塞缓解。

本文的创新工作如下。

1) 提出了面向多芯粒 LLM 解码推理的预算守恒带宽配置范式。在满足平台复用约束的前提下，突破了现有研究仅侧重需求侧任务映射或依赖高成本硬件重设计的局限，将逐链路带宽空间重分布纳入优化变量。通过从供给侧定向增强瓶颈截面，在不改变硬件设计的条件下解决了由强空间非均衡导致的链路过早饱和问题。该范式以极低的实现成本显著提升了既有硬件平台对不同模型规模与负载形态的可扩展性，并给出了优化过程的收敛性证明。

2) 建立了应对 LLM 负载解码流量动态特征的协同优化体系。引入了链路级等效突发因子建立尾部稳定性代理模型，实现了对突发性流量特征的量化感知识别，并提出了基于热点重合率的触发式更新机制，有效解决了解码阶段不可避免的瓶颈截面漂移难题。该体系通过配置参数与流量动态特征的精准适配，消除了静态配置难以覆盖运行态热点迁移的痛点，确保了离散通道配置过程在有限步内终

止并达到局部稳定。上述方法的有效性通过周期性精确仿真环境得到验证。

1 相关工作

1.1 模型与算子背景

现有研究沿算子、内存、服务3条路径推进推理优化。在算子层面，文献[8]指出注意力的主要瓶颈在于键值数据读写而非矩阵运算本身，面向I/O的注意力方法通过重排计算与访存顺序，减少高带宽存储与片上存储之间的往返，从而在不改变模型语义的前提下降低时延并提高能效。在内存层面，推理服务需在多并发请求与长上下文间共享有限显存及高带宽存储。分页化键值（KV）缓存方案通过固定大小页组织数据，结合虚拟寻址与页表映射，提升并发能力[10]。这类方法在兼顾地址转换开销的同时，使缓存命中率与批处理效率相互促进，对用户可见的响应时间带来稳定收益。在服务层面，文献[9]从系统调度角度区分预填充与解码阶段，并围绕服务级目标（service-level objective，SLO）设置优先级和配额，以减轻阶段间干扰，在高负载下维持有效吞吐量和尾部稳定。与此同时，存内和近内存计算通过在存储侧下沉部分算子或数据重排，在保证精度的前提下降低访存能耗并减轻平均带宽压力[11-12]。

整体来看，上述工作从算子重构、缓存管理和服务调度等角度优化LLM推理性能，为后文建模提供了流量模式和资源需求的基础刻画。但它们主要作用于单加速器或单节点视角，尚未在固定封装形态与D2D约束下针对封装内多芯粒网络的带宽布局展开专门设计。

1.2 互连与封装方法学

在后摩尔时代的制造与成本边界下，多芯粒封装成为扩展性能与容量的可行路径。与片上网络相比，封装级互连面临带宽密度与能效约束，易在热点链路形成排队并放大解码阶段的尾部风险。有研究提出以工作负载特征为输入，开展拓扑、路由与链路容量的协同设计，在既定封装与成本约束内进行可验证的折中与配置搜索[3,13]。从路线展望和产业实践看，芯粒时代的研究版图需要封装、互连和架构的一体化协同[14]。在互连媒介方面，可重构硅光在组播、广播与高基数扩展上展现潜力，可在关键路径集中带宽并降低远距离通信代价[4-5]。先进封装明确了可实现的面积与布线边界，面向多芯粒环境的安全互连接口强调在故障与异常场景下的冗余与仲裁能力[15-16]，以提升整体系统的鲁棒性和安全性。上述结果共同表明，仅关注带宽总量难以充分刻画NoP的性能行为，带宽在空间上的配置位置以及与封装、互连和安全机制的协同同样关键。

1.3 封装结构与带宽布局

任务映射与算子划分是影响跨芯粒通信的前置因素，相关研究表明，通过计算图空间分解与物理位置对齐可显著改善拥塞分布[17-18]。文献[17]进一步指出，针对少数关键链路的容量微调被证明能在不修改微结构的前提下获得可观收益。此外，结合批处理规模与预存取策略的调度方案有助于缓解因KV流量引起的互连拥塞[19-20]。这些工作从映射、调度与微结构等不同层次改进封装内数据通路，对封装结构和带宽布局下的性能上界给出了重要参考，但多以平均通信成本或整体扩展性为主要关注点，尚未在严格匹配算力、存储与封装带宽预算的约束下系统考察面向LLM解码阶段p99行为的带宽再配置策略。

1.4 特征分析与封装

面向端侧与低并发场景，文献[7]通过异构芯粒解耦算力与存储，减少封装内权重与KV搬运，在功耗与成本受限的情况下仍满足时延与能效需求。多芯粒映射与设计空间探索方面，有研究基于跨芯粒流水与依赖关系的图模型探索映射空间[21-23]。通过在映射和调度层面显式约束带宽分配侧重于数据驻留位置，缓解解码阶段的带宽瓶颈与负载不均[24]。随着一体化工具链的完善及3D/近存计算的演进[25-26]，跨芯粒资源编排日益精细化。面向通用扩展的芯粒化存算一体框架则为跨算子与数据通路的复用与扩展提供了新范式[27-28]。本文方法与现有研究的对比如表1所示。

相比于现有的多芯粒加速器研究，T²-CHIP在处理解码流量动态挑战方面具有显著优势。不同于Gemini[6]侧重平均负载均衡的静态映射工作，本文通过链路级等效突发因子κ_e建模与触发式更新机制，实现了对突发性、非均衡性与热点漂移的系统性抑制。这种以带宽弹性应对流量动态的范式，有效解决了既有研究在固定互连资源约束下难以稳定控制窗口化p99尾部时延的难题，为多芯粒LLM推理系统的平台级复用提供了可复现的优化路径。

表 1　　多芯粒加速器协同优化方法对比

优化维度	工作	核心优化目标	关键控制变量	互连建模特征	解码阶段突发/非均衡处理
计算/存储侧	TransPIM[11]/HAIMA[12]	降低访存功耗与时延	存内计算逻辑与算子重构	静态带宽或理想总线	侧重权重驻留与访存量
映射/需求侧	Gemini[6]/M2M[21]	提升平均吞吐量与利用率	算子切分模板与任务映射	固定带宽的网格/树形拓扑	侧重平均通信量优化
互连/建模侧	INDM[22]/Das[23]等	降低平均跳数与时延能量积	拓扑搜索与数据流映射	分析性时延模型	通用 DNN 建模，未考虑突发性
协同/供给侧	T²-CHIP	窗口化 p99 尾部稳定性	逐链路带宽重分配 + 角色对齐	周期精确模拟 + 突发因子建模	显式建模突发与热点漂移

2　设计与方法

在统一封装形态与资源预算的约束下，T²-CHIP 的核心问题是：在不改变 NoP 拓扑结构及路由器微架构的前提下，通过任务角色与芯粒能力对齐，并在总 D2D 带宽预算守恒条件下对芯粒间的链路容量进行空间重分布，以缓解解码阶段 NoP 热点排队并降低窗口化尾部时延，同时兼顾首 token 时延与整体吞吐量。

解码阶段 NoP 性能退化源于两类特征：第一类是 token 节拍引起的开关式突发，使窗口化尾时延对局部链路排队高度敏感；第二类是注意力交互带来的强空间非均衡，导致热点位置随负载形态漂移。为此 T²-CHIP 采用契约驱动的三阶段闭环：阶段 1 基于严格定义的合成生成器构造可复现输入，阶段 2 求解预算守恒下的连续容量分布，阶段 3 执行 SLO 驱动的离散化闭环校正，降低尾部偏差风险。

本文将阶段 1 输出且在阶段 2 与阶段 3 保持不变的一组参数定义为契约，包括固定封装形态、NoP 拓扑、路由策略、路由器微结构参数、观测窗口、负载缩放、流量类别集合、分层流量画像统计量、总带宽预算、链路容量下限与通道粒度。在契约约束下，阶段 1 仅在预定义的角色划分与映射模板集合内进行筛选与画像统计，阶段 2 与阶段 3 仅对逐链路容量配置进行优化与校正。本文所有对比方案均在同一契约与同一统计口径下计算窗口化尾部指标与链路利用率指标。

本文关注解码阶段突发与非均衡流量在 NoP 中引起的排队与拥塞现象，并由此导致窗口化尾部时延与最忙链路利用率变化。为保证因果归因清晰，仿真中显式建模与排队强相关的网络机制，包括 wormhole 传输与排队、基于 credit 的背压、虚通道与缓冲管理、交换与仲裁分配、逐跳流水线时延、链路序列化时延，以及由阶段 1 合成生成器产生的突发注入流量轨迹（trace）。为隔离优化变量的影响，本文在不同方案间保持 NoP 拓扑、路由策略与路由器和物理层微结构参数一致。

2.5D 多芯粒架构与 T²-CHIP 的作用位置如图 1 所示。上层为解码计算结构，基于子模块计算特征差异，本文将芯粒划分为两类角色：X 类芯粒侧重承载注意力与键值数据相关交互，优先配置 NoP 端口与外部存储通道；Y 类芯粒侧重承载前馈网络计算与片上复用，优先配置算力资源。中间层为异构多芯粒封装与封装内网络。下层展示封装内物理互连结构。系统在给定映射与负载下产生的解码流量经统计聚合形成分层流量画像，用作容量配置优化的输入。容量配置优化输出逐链路带宽配置并作用于 NoP，以缓解关键截面的热点拥塞。

本文采用契约驱动的三阶段工作流程，如图 2 所示。阶段 1 在逻辑侧进行快速设计空间探索（design space exploration，DSE）与分层流量画像，在若干 X/Y 芯粒角色配置及其对应的映射模板上，利用 Roofline 模型筛除显失均衡的候选方案，并统计固定观测窗口 W 内各类流量的平均承载率、窗口化 p99、分组大小及突发特性。阶段 2 在既定分层流量画像（layered traffic profile，LTP）的基础上，将容量配置优化建模为带约束的非线性规划，求解总带宽 B 不变下的连续解，并将其离散化为 SerDes 通道粒度的配置。阶段 3 在周期精确仿真中，执行 SLO 驱动的闭环校正，通过单通道成对交换进行有限次微调，得到通道粒度的可实现容量配置。

解码阶段的跨芯粒通信具有突发性与强非均衡性，流量容易在拓扑中聚集形成热点排队，使尾部时延对局部链路容量高度敏感。本文在流量画像中

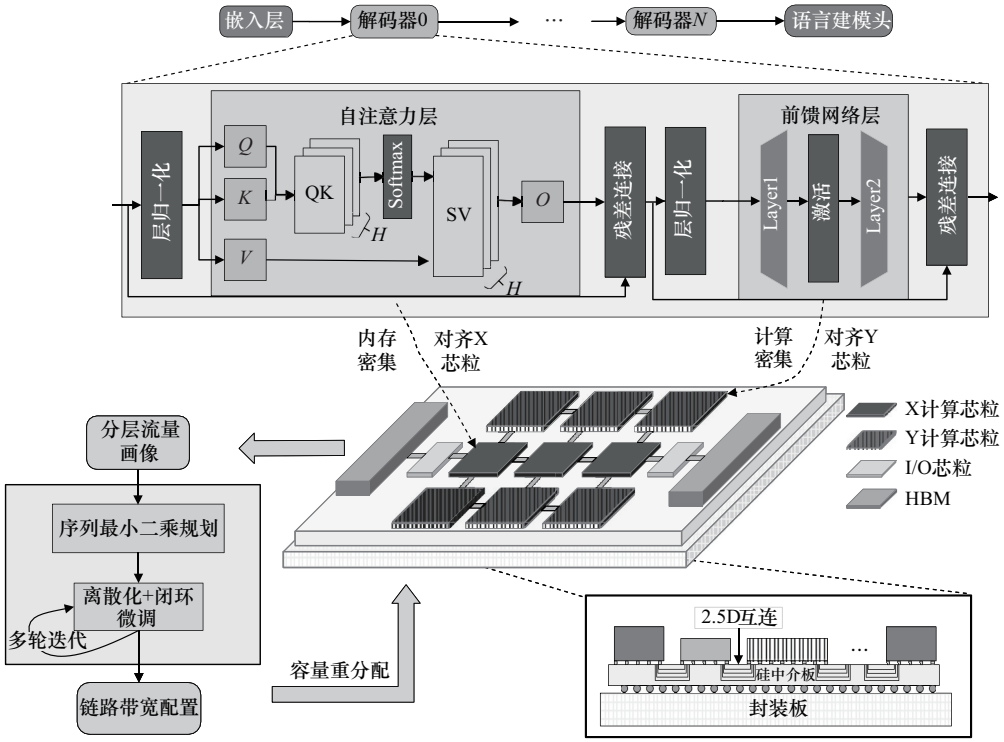


图 1　2.5D 多芯粒架构与 T²-CHIP 的作用位置

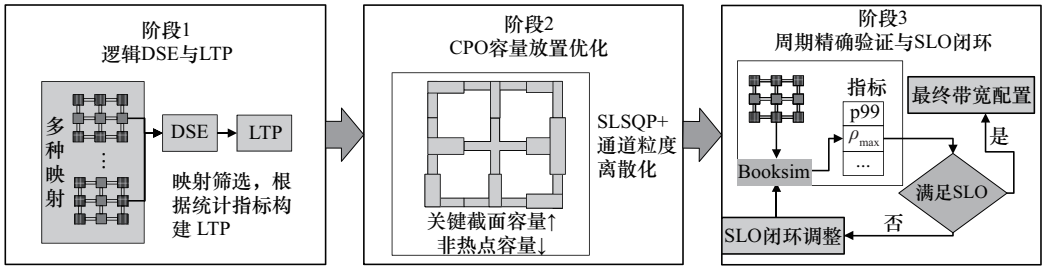


图 2　T²-CHIP 三阶段流程

同时刻画平均承载率与窗口化 p99 承载率，引入占空比与突发因子表征 token 节拍引起的时序波动，并据此识别窗口化峰值压力显著的热点链路，定位瓶颈截面。本文以链路 e 在各观测窗口内的平均承载率序列为基础，记其均值为 λ_e，并对该序列取第 99 百分位得到 λ_e^{p99}，按 λ_e^{p99} 降序排序取前 10 条链路构成热点链路集合，并据此刻画瓶颈截面的空间位置。在闭环校正阶段，以窗口化尾部指标与最忙链路利用率为约束目标，对通道粒度配置进行有限次调整，以降低突发条件下仅依赖连续解带来的尾部偏差风险。

为使阶段 2 的尾时延解析代理对突发性具备可复现且一致的输入口径，在链路尺度上定义等效突发因子。记链路 e 的窗口化平均承载率序列的均值为 λ_e，第 99 百分位为 λ_e^{p99}，则定义

$$\kappa_e \triangleq \frac{\lambda_e^{\mathrm{p99}}}{\lambda_e} \tag{1}$$

为链路在窗口尺度上的峰均比，由阶段 1 窗口化统计唯一确定，并作为阶段 2 排队项的突发放大建模。阶段 1 的突发因子 κ 用于描述生成器输入的开关式时序结构，κ_e 则是在链路尺度上叠加多流并引入峰值窗口机制后的统计等价量，在其他条件固定的情况下，增大 κ 会使窗口内注入更集中，从而使 κ_e 具有同向增大的趋势。当仅改变全局负载缩放 θ 时，链路平均承载率与窗口化 p99 承载率按比例缩放，κ_e 保持不变，因此可以将突发结构与平均负载水平解耦。

瓶颈截面会随模型规模、上下文长度与负载水平变化而迁移，导致单一配置难以覆盖所有服务形态。当热点链路集合的重合率显著下降时，需要更

新封装内链路的通道粒度容量配置，而不改变芯粒与互连微结构，重合率的定义与阈值设置见3.4节。工程实现中可面向典型服务形态离线生成少量容量配置档位以覆盖常见负载区间，兼顾配置更新成本与适配效果。本文主要符号如表2所示。

表2　　本文主要符号

符号	名称与含义
W	LTP 观测窗口长度/ms
θ	负载缩放因子
D	占空比（ON比例）
κ	突发因子（流级ON/OFF，近似$\kappa \approx 1/D$）
κ_e	链路e的等效突发因子
$\bar{s}$	平均分组大小/B
s_f	流f的平均分组大小/B
E	链路集合
λ_e	链路e的平均承载率/(GB·S^{-1})
C_e	链路e的容量/(GB·S^{-1})
B	封装级D2D总预算/(GB·S^{-1})
$C_{\min}$	链路容量下限/(GB·S^{-1})
ρ_e	链路e的利用率
$\rho_{\max}$	最忙链路利用率
$C_{s,e}^2$	服务时间变异系数平方
γ	代理校准因子
L_f^{p99}	流f的p99时延代理/ms
α	目标权重（p99与$\rho_{\max}$折中）
ϕ	算力倾斜系数
ΔCch	SerDes通道粒度/(GB·S^{-1})
ρ_{tar}	最忙链路利用率阈值
τ_{low}	低利用率阈值

为保证评估可复现，本文在统一契约约束下开展后续优化与验证，并且后续阶段均在相同固定项下进行，仅针对逐链路容量配置进行优化与校正。

2.1　角色对齐异构性与系统模型

在T²-CHIP中，系统采用与Transformer子模块特性对齐的异构芯粒划分，将芯粒分为X和Y两类。X芯粒面向注意力与键值数据的进出，优先配置NoP端口与外部存储通道；Y芯粒主要承担前馈网络计算与片上复用，侧重提升算力密度与权重复用能力。在典型Transformer配置下，按每秒浮点运算次数（FLOPS）估算FFN，相关计算量约占层内总量的60%~70%，据此将总算力中的$\phi \approx 0.67$分配给Y芯粒，其余分配给X芯粒。此分配反映了FFN的计算主导性，但不改变系统的总FLOPS。NoP采用固定坐标的网格化拓扑，虚通道数、缓冲深度等在不同系统间保持一致，以隔离容量配置的影响。

优化变量并非拓扑，而是芯粒间链路e的容量C_e（单位为Gbit/s）。记E为NoP的有向链路集合，总D2D带宽为B，则总带宽守恒关系为

$$\sum_{e \in E} C_e = B \tag{2}$$

该约束意味着容量再配置是零和博弈：向热点链路区域增加容量，必然伴随着对非热点区域的等额回收。为避免引入路由策略这一额外变量，本文默认采用最短路径映射，当存在多条等长路径时，采用等价最短路径分摊进行流量划分。

2.2　阶段1：逻辑DSE与分层流量画像

在昂贵的精细仿真与大规模搜索之前，先用轻量模型筛除显失均衡的映射与角色划分，可有效避免低效的盲目探索。为此，本文对多种X和Y分工与层级映射模板进行快速评估，采用屋顶线（Roofline）模型得到的瓶颈上界来近似阶段时间。设$\varphi \in \{\text{prefill,decode}\}$，则阶段时间$T_\varphi$上界可定义为

$$T_\varphi = \max\left\{T_{\mathrm{comp}}, T_{\mathrm{mem}}, T_{\mathrm{comm}}\right\} \tag{3}$$

其中，T_{comp}、T_{mem}、T_{comm}分别为计算、存储、通信主导的耗时估计。与此同时，基于LTP的平均负载投影定义链路承载率与链路利用率，记λ_e为根据LTP在观测窗口内统计得到的链路e的平均字节率，即所有数据流在该链路上的平均承载之和，则链路利用率与最忙链路利用率分别定义为

$$\rho_e = \frac{\lambda_e}{C_e},\ \rho_{\max} = \max_{e \in E} \rho_e \tag{4}$$

当候选组出现接近饱和的链路，即$\rho_{\max}$过高时，直接剔除该组以减轻后续搜索负担。对于通过筛选的映射，进一步生成分层流量画像的参数集合作为后续优化与验证的契约，并由该参数集合驱动一个严格定义的合成生成器，用于产生可复现的周期精确仿真输入。在固定观测窗口W上，LTP以源–宿–类别三元组对流进行分层，其中类别包含KV、QK、AV、FFN等，其中QK表示Query-Key相似度计算，AV表示对Value的加权聚合。对于每一层，记录3类关键统计量：平均字节率与窗口化

p99 字节率（单位为 Gbit/s），分别刻画稳态负载与峰值负载；平均分组大小 $\bar{s}$，影响序列化时延；占空比 D 与突发因子 κ（用 $\kappa = \frac{1}{D}$ 近似）。由于解码按 token 节拍运行，数据呈现出交替的活跃与空闲时序结构，上述统计可显式反映该节拍性。

观测窗口 W 决定了 LTP 对短时突发的刻画尺度，W 过大可能稀释突发强度，W 过小则可能放大随机抖动并影响统计稳定性。本文将 W 作为契约的一部分并在阶段 2、阶段 3 保持一致，以确保容量配置的对比可复现。不同 W 下的敏感性在 3.5 节进行讨论。

为直观展示窗口统计与开关节奏，LTP 的窗口化建模如图 3 所示，其中，LTP 时间窗口 W= 64 ms，解码突发 D≈0.15，$\kappa = \frac{1}{D}$。ON/OFF 节拍将解码阶段的短时突发保留在观测窗口内，避免时间平均掩盖尾部压力。此外，引入全局负载缩放因子 θ，用于按比例放大或缩小 LTP 以扫描不同的负载点。LTP 的优势在于：一方面，不对解码阶段的同步流量进行时间平均，而是在窗口 W 内以接近 token 的时间分辨率保存短时突发与细粒度行为；另一方面，表征与物理几何解耦，便于在固定网格与统一路由的假设下开展可重复的优化与仿真。

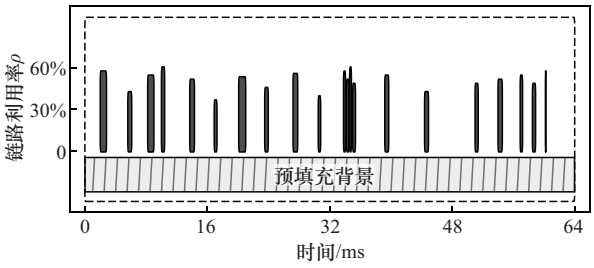


图3 LTP 的窗口化建模

本文使用了严格定义的合成注入生成器来实现阶段 1 的 LTP 画像。生成器基于观测窗口 W，对于每个源–宿–类别三元组流 f=(src,dst,type)，生成固定长度为 N_w 的注入序列，并确保该序列与 LTP 中的画像参数一致。生成器的输入参数包括窗口内平均字节率 $\bar{\lambda}_f$、窗口化 p99 字节率 λ_f^{p99}、分组大小 s_f、占空比 D_f 和全局负载缩放因子 θ。

生成器通过以下 3 个步骤来保证注入流量的准确性和可复现性。

1) 窗口强度分配：根据占空比 D 和 p99 值调整窗口的峰值强度，确保生成的流量符合统计特性。

2) ON/OFF 区间划分：将每个窗口划分为 ON 和 OFF 区间，模拟突发流量。

3) 事件级到达展开：仅在 ON 区间内生成分组到达事件，并均匀分布到达时刻。

通过这些步骤，生成器能够精确地模拟解码阶段的突发性和非均衡性。

2.3 阶段 2：容量配置优化

在固定网格与既定 LTP 的条件下，容量配置的优劣直接体现在两类指标上：一是网络是否逼近拥塞边界，以最忙链路利用率 $\rho_{\max}$ 衡量；二是服务侧最关心的窗口化 p99 尾时延。前者反映结构与余量，后者反映用户体验。

为刻画尾时延，本文将链路抽象为 $G/G/1$ 队列，并对关键流 f 沿其路径 path(f) 给出窗口化 p99 的解析代理，即

$$L_f^{\mathrm{p99}} \approx \sum_{e \in \mathrm{path}(f)} \left(\frac{\bar{s}_f}{C_e} + \frac{\bar{s}_f}{C_e} \cdot \frac{\rho_e}{1-\rho_e} \cdot \kappa_e + \frac{C_{s,e}^2}{2} \cdot \gamma \right) \tag{5}$$

其中，第一项为序列化时延；第二项为排队时延代理，体现利用率、到达与服务方差以及服务时间的共同作用；第三项为服务时间变异项，用于刻画服务时间随机性及模型近似误差对尾时延的影响。针对解码流量的突发性与非均衡性，式 (5) 的核心物理意义在于引入等效突发因子 κ_e 能够量化 token 生成节拍引起的瞬时排队压力，使容量配置优化不再仅基于平均带宽，而是能够自动识别并向高负载、强非均衡的热点链路倾斜资源，从根源上抑制瓶颈截面的形成。阶段 3 的闭环校正则进一步解决瓶颈随负载形态漂移导致的静态配置失效问题。系统级尾时延代理取关键流的最大值

$$\mathrm{p99} = \max_f L_f^{\mathrm{p99}} \tag{6}$$

在制造可行的前提下，要求每条链路容量不低于下限 $C_{\min}$（单位为 Gbit/s），以维持连通性与最小可路由性。将容量配置写成带等式与不等式约束的非线性规划，令 α 为任务优先级权重，有

$$\begin{aligned} \min_{\{C_e\}} \quad & \alpha \cdot \frac{\mathrm{p99}(\{C_e\})}{\mathrm{p99}_{\mathrm{ref}}} + (1-\alpha) \cdot \frac{\rho_{\max}(\{C_e\})}{\rho_{\mathrm{ref}}} \\ \text{s.t.} \quad & \sum C_e = B,\ C_e \geqslant C_{\min},\ \forall e \in E \end{aligned} \tag{7}$$

在固定拓扑与总带宽预算下，容量配置的优劣主要体现在两类指标上：窗口化尾部时延反映服务

稳定性，最忙链路利用率反映网络逼近拥塞边界的程度。为同时抑制尾部抬升并避免将系统推向饱和区域，式(7)对两项指标进行归一化后加权，形成目标用于阶段 2 的快速搜索。阶段 3 在周期精确仿真中直接度量窗口化 p99 与 $\rho_{\max}$，对通道粒度配置进行闭环校正。其中，$\mathrm{p99}_{\mathrm{ref}}$ 与 ρ_{ref} 为归一化参考量，本文取 $\mathrm{p99}_{\mathrm{ref}}$ 为基线方案在相同观测窗口与负载缩放下得到的窗口化 p99，ρ_{ref} 为目标利用率阈值 ρ_{tar}。因此式(7)中的两项分别表示相对基线尾时延与相对拥塞程度，α 为控制二者的折中权重。

求解步骤分两步。第一步，在容量连续空间中求解式(7)，采用序列最小二次规划（sequential least quadratic programming，SLSQP）方法，该方法属于序列二次规划（SQP）方法的一种，适用于处理带有等式与不等式约束的光滑目标优化问题。初始化使用均匀分配，在其基础上叠加多次小幅随机扰动以降低局部次优风险，得到的 $\{C_e\}$ 为连续解。

为补充说明阶段 2 连续优化的求解性质，下面给出在标准假设下采用 SLSQP 的局部收敛结论。

命题 1　阶段 2 连续容量配置在可行域内具有局部收敛性。

证明　在契约固定条件下，式(7)定义了带等式约束与不等式约束的非线性规划。若目标函数与约束函数在可行域内连续可微，并满足标准的约束资格条件，且可行域非空，则采用序列二次规划方法求解时，其迭代序列在常见假设下可收敛到满足一阶最优性条件的局部驻点。由于该问题一般为非凸优化，局部驻点不保证为全局最优。本文通过多次扰动初始化降低局部次优风险，并在阶段 3 引入周期精确指标驱动的通道粒度闭环校正，使最终配置在部署粒度与系统级指标上更稳定。证毕。

在获得连续域解后，施加封装与 RDL 的离散化约束，并将其量化到 SerDes 通道粒度进行预算守恒下的微调。

第二步，施加封装与 RDL 的离散化约束。链路带宽以 SerDes 通道为最小配置单位，容量取值属于离散的可实现集合，先将连续解就近映射为对应的通道数，再在固定总带宽约束下进行成对交换的贪心微调，每一步在一条链路上增加一个通道，同时在另一条链路上减少一个通道，优先调整对目标函数贡献更大的链路，直至满足 $\sum C_e = B$，且目标不再下降。当需要加快早期收敛时，可选用基于链路线长和线宽的轻量预分配作为起点。

由式(5)和式(6)给出的窗口化 p99 解析代理来源于队列近似，它能够较好地反映容量变化带来的相对趋势，但在绝对数值上可能整体偏大或偏小。为降低这种整体尺度误差，本文在基线容量分配下，采用与后续优化一致的观测窗口 W 与负载缩放 θ，分别计算解析代理与周期精确仿真得到的真实窗口化 p99，并定义校准因子为

$$\gamma = \frac{\mathrm{p99}_{\mathrm{sim}}}{\mathrm{p99}_{\mathrm{proxy}}} \tag{8}$$

随后在阶段 2 中对解析代理进行统一缩放，使其在该参考点与周期精确仿真对齐。需要说明的是，γ 仅用于修正整体尺度偏差，无法完全刻画不同容量配置下的排队非线性变化，因此阶段 3 仍需在周期精确仿真中对通道粒度配置进行闭环校正。

2.4　阶段 3:周期精确验证与 SLO 闭环

阶段 3 以阶段 2 输出的通道粒度容量配置 $\{C_e\}$ 为起点，在固定总带宽预算与固定拓扑、路由条件下进行周期精确仿真，直接度量解码阶段窗口化 p99 以及最忙链路利用率 $\rho_{\max}$。当指标未达到预设目标时，采用等带宽单通道成对交换在通道粒度上进行有限次校正。该闭环不仅用于修正连续解的量化偏差，也用于应对强空间非均衡导致的排队非线性放大。闭环每轮迭代根据当前仿真得到的链路利用率集合 $\{\rho_e\}$ 构造接收候选集合 H，H 由利用率最高的部分链路构成，供给候选集合 L_{set} 由利用率较低且容量高于下限的链路构成。每次迁移从 L_{set} 中选取一条链路减少一个通道，并从 H 中选取一条链路增加一个通道，从而在预算守恒的条件下将通道资源定向配置到拥塞更敏感的链路上。为对候选迁移进行排序，本文定义迁移评分 L，并仅接受能够使 L 下降的迁移。停止条件与可行性判断以周期精确仿真的窗口化 p99 与 $\rho_{\max}$ 为准。仿真流量由阶段 1 合成生成器生成，观测窗口 W 和负载缩放 θ 与阶段 2 保持一致，从而保证闭环校正过程可复现。具体步骤如算法 1 所示。

算法 1　等带宽闭环校正

输入　拓扑与路由、LTP 统计、观测窗口 W、负载缩放 θ、随机种子、初始容量 $\{C_e\}^0$（阶段 2 输出）、ρ_{tar}（SLO 中设定的最忙链路利用率阈值）、低利用阈值 τ_{low}、最大迭代轮数 $K_{\max}$、权重 α、容量下限 $C_{\min}$、单通道带宽 ΔCch、收敛容差 ε

输出 最终容量$\{C_e\}^*$以及对应的p99与$\rho_{\max}$

1) 用$\{C_e\}^0$仿真，得到$\{\rho_e\}$、p99、$\rho_{\max}$，按式(7)计算当前迁移评分L。
2) for j = 1:1:$K_{\max}$ do
3) 构造接收集合H：按ρ_e降序，取利用率最高的前5%链路。
4) 构造供给集合L_{set}：满足$\rho_e \leqslant \tau_{\text{low}}$且$C_e \geqslant C_{\min}+\Delta\text{Cch}$的链路，并去除$H$中的元素。
5) if H为空 or L_{set}为空 then 输出$\{C_e\}$、p99、$\rho_{\max}$，终止。
6) 设$L_{\text{best}} \leftarrow +\infty$，best_pair未定义，同步预留$\text{p99}_{\text{best}}$与$\rho_{\max}^{\text{best}}$。
7) for 每个$e^+ \in H$ do
8) for 每个$e^- \in L_{\text{set}}$ do
9) 临时迁移：$C_{e^+} \leftarrow C_{e^+}+\Delta\text{Cch}$，$C_{e^-} \leftarrow C_{e^-}-\Delta\text{Cch}$。
10) 仿真得到$\text{p99}_{\text{trial}}$、$\rho_{\max,\text{trail}}$，按式(7)计算$L_{\text{trial}}$。
11) 恢复C_{e^+}、C_{e^-}到迁移前取值。
12) if $L_{\text{trial}} < L_{\text{best}}$ then
13) $L_{\text{best}} \leftarrow L_{\text{trial}}$，best_pair $\leftarrow (e^+,e^-)$。
14) $\text{p99}_{\text{best}} \leftarrow \text{p99}_{\text{trial}}$，$\rho_{\max}^{\text{best}} \leftarrow \rho_{\max}^{\text{trail}}$。
15) end for
16) end for
17) if $L_{\text{best}} \geqslant L-\varepsilon$ then
18) 输出$\{C_e\}$、p99、$\rho_{\max}$，终止。若评分无法下降且$\rho_{\max} > \rho_{\text{tar}}$或p99仍未达到目标，则记为局部最优。
19) 执行真实迁移：对best_pair的e^+增加ΔCch，对e^-减少ΔCch。
20) 仿真并更新$\{\rho_e\}$、p99、$\rho_{\max}$；按式(7)更新L。
21) end for
22) 输出$\{C_e\}$、p99、$\rho_{\max}$。

为了刻画阶段3闭环校正的收敛性和搜索过程的可行性，给出命题2。

命题2 阶段3闭环校正在有限步内终止，并得到候选邻域内的局部稳定配置。

证明 令链路e的通道数为$n_e = \dfrac{C_e}{\Delta\text{Cch}}$，在总带宽预算守恒与链路容量下限的约束下，可行的通道数向量集合为有限集合。算法1每次真实迁移仅对两条链路的通道数作等量增减，并要求迁移评分严格下降，因此不可能在有限集合上形成无限长的严格下降序列，算法1必在有限步内终止。算法1终止时，在其构造的候选集合H与L_{set}所枚举的全部单通道成对交换中，不存在能使评分下降超过容差ε的迁移，因此当前配置在该候选邻域内达到局部稳定。证毕。

因此，阶段3能够在预算守恒和容量下限约束下，通过有限次迁移得到稳定的通道粒度配置，并保证在局部邻域内部署。

为进一步保证可评估与可复现，本文将评估流程抽象为契约驱动的简化系统模型，如图4所示。该模型以契约参数与映射和角色设定为输入：阶段1在固定观测窗口与负载缩放下生成可复现的注入trace；阶段2在总带宽预算守恒与链路容量下限约束下求得逐链路容量连续解，并将其量化为通道粒度可实现配置；阶段3在周期精确网络仿真口径下对通道粒度配置进行服务等级目标驱动的闭环校正，输出最终可部署配置。所有对比方案在相同简化模型与相同统计口径下计算窗口化p99、最忙链路利用率与端到端指标，从而保证差异归因于映射与逐链路容量分布，而不是拓扑、路由与微结构变化。

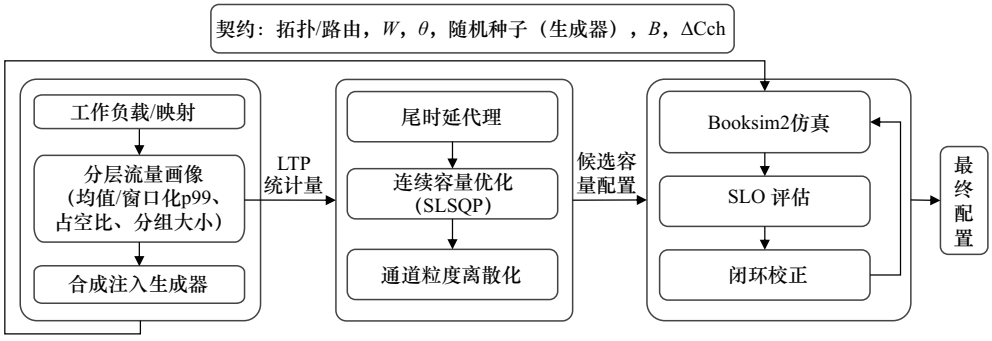


图4 契约驱动的评估简化系统模型与闭环流程

2.5　复杂度分析

T²-CHIP 的三阶段流程在计算开销上呈前轻后重的特点，阶段 1 与阶段 2 用于快速筛选与连续空间求解，阶段 3 以周期精确仿真作为主导开销来源。下面分别给出各阶段的主要复杂度来源。

阶段 1 在给定若干组角色划分与映射模板的候选集合上进行筛选与画像统计。设候选模板数量为 N_{tpl}，NoP 有向链路数为 $|E|$，在固定观测窗口 W 与流量类别集合下，LTP 统计主要由对每条链路的窗口统计与分层聚合构成，其时间开销与链路数及候选数量近似成正比，复杂度为 $O(N_{\mathrm{tpl}}|E|)$。该阶段不涉及周期精确仿真，开销可控，主要用于减少后续阶段的无效搜索空间。

阶段 2 采用 SLSQP 在连续空间求解式(7)。设 SLSQP 的迭代次数为 T_{opt}，每次迭代需要评估约束及目标：约束评估主要为总带宽守恒与逐链路下界检查，目标评估需要计算关键流的 p99 与 $\rho_{\max}$，其开销与链路数及关键流路径总长度相关。令关键流集合为 F，则阶段 2 连续求解的复杂度可近似为 $O(T_{\mathrm{opt}}(|E|+\sum_{f\in F}|\mathrm{path}(f)|))$。连续解离散到 SerDes 通道粒度后，采用成对交换的贪心微调。设离散化后的交换尝试次数为 T_{swap}，每次交换需要增量评估目标变化，其开销同样与受影响链路及相关路径有关，整体可近似为与 $|E|$ 成线性关系，因而阶段 2 整体随网络规模可扩展。

阶段 3 以周期精确仿真为核心开销。设最大迭代轮数为 K，每轮迭代中，需要从接收集合 H 与供给集合 L 中枚举候选迁移对，并对候选对进行仿真评估以计算迁移评分。令每次周期精确仿真的代价为 T_{sim}，则阶段 3 的最坏复杂度为 $O(K|H||L|T_{\mathrm{sim}})$。

在实现中，H 由最忙链路的前 5% 构成，L 由低利用率阈值筛选得到，二者规模均由超参数 k 与阈值控制，从而将候选枚举限制在可控范围内，使闭环校正在有限轮次内完成。

综合来看，阶段 1 与阶段 2 的计算代价主要随链路数近似线性增长；阶段 3 由周期精确仿真主导，代价与仿真轮数及候选迁移对规模成正比，因而其开销可通过候选比例与阈值策略进行控制。

2.6　局限性与适用范围

为保证对比的可复现性并突出容量配置的作用，本文在模型、约束与方法上进行了若干设定，这也带来了相应的适用范围与局限性。

1) 固定拓扑与路由器微结构的前提

本文在契约中固定封装内网络拓扑、路由策略与路由器微结构参数，T²-CHIP 关注在既定封装形态与互连微结构下，通过逐链路容量重分布与角色对齐改善拥塞结构与尾部指标。本文方法不适用于回答拓扑设计与路由器设计的最优性问题。

2) 面向服务形态的契约一致性

分层流量画像刻画的是给定服务形态与负载区间下的解码阶段通信特征，并作为阶段 2 与阶段 3 的共同输入以保证统计口径一致。当模型结构、上下文长度、并发区间或调度策略发生显著变化时，热点链路与瓶颈截面可能迁移。此时，需要更新的是契约中与流量画像相关的部分，并重新生成通道粒度容量配置，而不是改变芯粒与互连微结构。实践中可针对典型服务形态离线生成少量容量配置档位，以覆盖常见负载区间，并降低频繁更新的工程成本。

3) 离散化与局部搜索的最优性边界

阶段 2 将连续解离散到 SerDes 通道粒度，并采用成对交换进行局部微调。阶段 3 采用等带宽单通道成对交换进行闭环校正。上述策略能够在有限步内得到满足约束且在候选邻域内局部稳定的配置，但不保证全局最优。该设定用于控制搜索代价并获得可复现的配置生成流程。

3　评估

3.1　实验设置

在相同的封装网格拓扑与芯粒坐标设定下，本文对同构网格网络（简称同构网格）、Gemini[6] 与 T²-CHIP 这 3 种方案进行对比评估。3 种方案在系统总量预算上严格对齐，即总算力、HBM 容量与带宽以及封装内总 D2D 带宽预算完全一致，同时固定 NoP 拓扑与路由、路由器微结构、链路序列化口径与逐跳流水线时延等参数。3 种方案的差异仅体现在两类决策上：第一类为任务放置与映射策略，第二类为逐链路容量配置 $\{C_e\}$。其中，同构网格作为基线方案采用均分映射，Gemini 采用其公开工作的映射策略，T²-CHIP 在任务放置与映射中引入角色对齐。

本文选取 LLaMA 系列模型的代表性规模与上下文长度组合，以覆盖解码主导、长上下文与中高并发等场景。对于每个工作负载点，首先依据模型

结构与映射结果统计解码阶段跨芯粒通信在源–宿对上的字节量与类别构成，并据此构造解码阶段的分层流量统计LTP。然后采用LTP的窗口模型在观测窗口W=64 ms内生成具有开关式突发特性的源–宿对注入序列，作为后续模拟器的输入，其中全局负载因子θ用于按比例缩放到达强度，从而在相同通信语义下扫描不同拥塞程度的运行点。

本文使用BookSim2[28]进行周期精确仿真，配置文件显式指定拓扑、路由、虚通道数等关键参数如表3所示。除非另有说明，每个负载点选取3组固定随机种子，对3种方案分别使用同一组种子独立运行3次，丢弃前5个预热窗口，并在随后20个窗口上统计指标。为确保稳态测量，本文记录预热阶段$\rho_{\max}$的收敛轨迹，并复核总预算与线长到流水线时延映射的一致性。本文采用周期精确网络仿真验证NoP拥塞与排队对窗口化尾部指标的影响。仿真建模边界与简化系统模型已在第2节给出，本节不再重复。为便于复现，本节重点说明BookSim2的关键配置参数、运行方式与统计流程。

表3 BookSim2 仿真关键参数

系统模块	配置
路由器微结构	router=iq；credit_based； num_vcs=8；vc_buf_size=8 flits； sw_allocator=islip，alloc_iters=1； buffer_policy=private；vc_allocator=islip input/output/internal speedup=1；include_queuing=1;
NoP拓扑与路由	3×3 2D mesh（k=3, n=2, c=1），subnets=1； routing_function=dim_order； use_noc_latency=1
逐跳流水线时延	routing_delay=1，vc_alloc_delay=1，sw_alloc_delay=1； st_final_delay=1，credit_delay=1
flit/字节口径	channel_width=128 bit（16 B/flit）

核心网络指标包括解码阶段窗口化p99尾时延及链路利用率相关指标，后者包括最忙链路利用率$\rho_{\max}$、Top-20链路平均利用率与链路利用率累积分布$F(\rho)$。端到端指标包括TTFT p99、TPS（token per second）以及TBT（time between token，用作TPOT的度量）。其中，本文TTFT采用解码阶段口径，即解码阶段首个token的时间，不含预填充阶段。

为保证可比性，Gemini基线遵循公开工作的协同优化设定，在评估时冻结路由器微结构与系统总量预算，并将其映射结果投影到本文统一的网格拓扑与线长配置上，从而避免拓扑或路由差异对结果的干扰。

本文主要评估7B/8k、7B/32k和13B/32k这3种规模，其中，7B/32k表示参数规模为70亿、最大上下文长度为32000 token的LLM。在主要目标运行点（本文默认采用7B/32k、并发32、θ=1.0），服务等级目标设为：解码阶段窗口化p99不超过同构网格的85%，且最忙链路利用率$\rho_{\max}$不超过80%。对于其他模型规模与上下文组合，80%用作拥塞健康阈值，用于比较不同方案在更重负载下的拥塞风险与稳定性。在系统级预算匹配的前提下，3种方案的配置差异如表4所示。

表4 各方案的配置差异

方案	计算资源分配	互连/映射
同构网格	均分	同构网格
Gemini	均分	Gemini映射
T²-CHIP	角色对齐	CPO

除非另有说明，容量配置和闭环校正阶段使用统一的优化参数设置：权重α=0.7，用于在解码阶段窗口化p99尾时延与$\rho_{\max}$之间折中。闭环校正中，接收侧候选链路取利用率最高的前5%链路，供给侧候选链路取利用率较低且容量高于下限的链路集合，一对迁移的目标是将$\rho_{\max}$压制在SLO中给定的阈值ρ_{tar}，以一个SerDes通道为步长执行一对迁移，并设置最大迭代轮数。

本文在不同阶段使用3类不同的Top集合：阶段1使用Top-10用于瓶颈截面定位与跨服务形态漂移量化，侧重解释性；阶段3闭环校正使用Top-5%构造接收集合H，旨在控制搜索规模并精准定位最敏感拥塞点；评估部分即本文第3节采用Top-20链路平均利用率作为统计量，用于度量不同负载下热点扁平化的程度。三者分别对应定位、校正与度量，统计口径不同但相互补充。

3.2　总体结果与端到端影响

在7B/8k、7B/32k、13B/32k这3种规模下，3种方案呈稳定且一致的排序，其中，同构网格尾时延最高，Gemini次之，T²-CHIP最低，如图5(a)所示。当规模与上下文增加时，尾时延增加，且不同方案差距加大。以7B/32k为例，T²-CHIP的解码阶段窗口化p99尾时延相对同构网格降低25.8%，同一运行点下，TTFT p99下降、TPS提升，端到端收益与网络侧尾部改善相吻合，如图5(b)所示，其中，柱状图对应左纵坐标，图形对应右纵坐标。

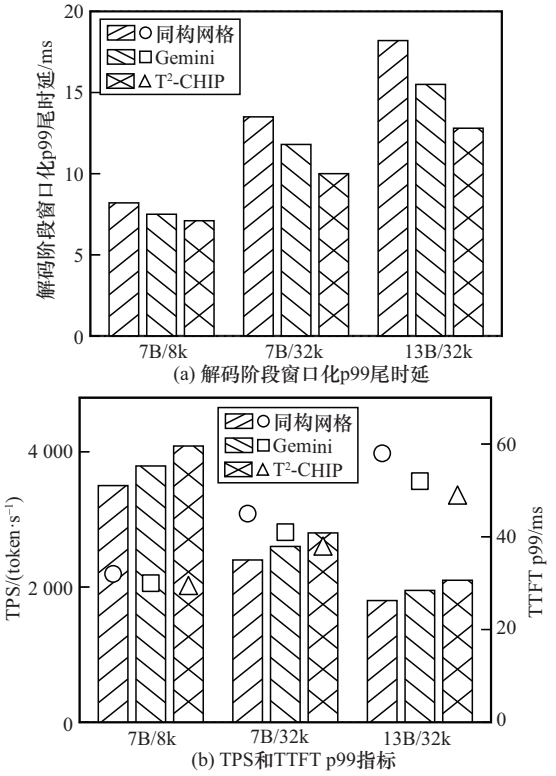


图5　目标负载场景与匹配预算约束下的性能评估结果

在总TOPS、HBM与总D2D预算完全匹配的条件下，上述结果表明，通过角色对齐与容量配置将有效NoP带宽集中于注意力/KV相关交互驱动的瓶颈截面，能够同步改善首token时延与吞吐率，且在长上下文场景中优势更明显。主要运行点的网络与端到端指标如表5所示，以7B/32k、并发32、θ=1.0为目标运行点，与同构网格相比，T²-CHIP的解码阶段窗口化p99尾时延由12.0 ms降至8.9 ms，下降了25.8%；TTFT p99由45.0 ms降至38.0 ms，下降了15.6%；TPS由2 400 token/s提升至2 800 token/s，提升了16.7%。

表5　主要运行点的网络与端到端指标

指标	同构网格	Gemini	T²-CHIP
解码阶段窗口化p99尾时延/ms	12.0	10.3	8.9
TTFT p99/ms	45.0	41.0	38.0
TBT p50/ms	7.5	7.0	6.3
TBT p95/ms	10.9	9.6	8.5
TBT p99/ms	13.2	11.5	9.8
TPS (token·s^{-1})	2 400	2 600	2 800
ρ_{max}	0.83	0.80	0.77
SLO是否满足	—	否（p99未达标）	是

端到端性能分量分解如图6所示。图6基于表5中的同一目标运行点，可进一步说明收益来源，柱顶数字为该运行点的单token p99尾时延，总量与图5(a)一致，柱体内标注的百分数表示网络分量在该总p99尾时延中所占的比例，仅当占比不低于30%时显示。以7B/32k为例，同构网格的网络占比约为40%，T²-CHIP降至约30%，计算与KV分量基本平稳，说明收益主要来自网络侧拥塞缓解与带宽优化，而非计算或KV的变化。

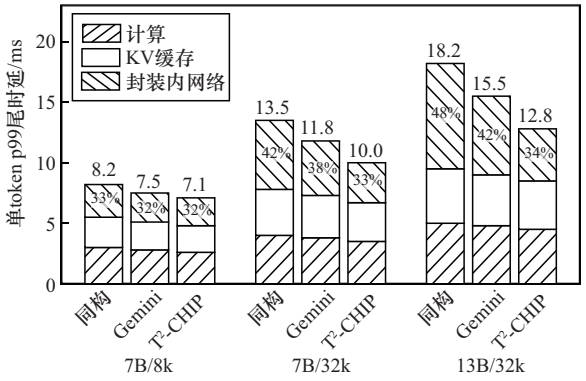
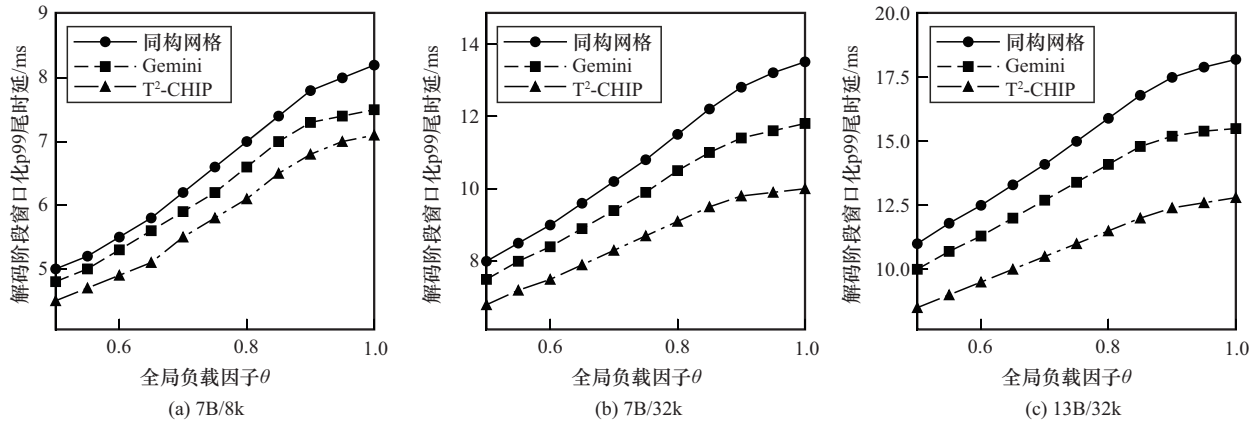


图6　端到端性能分量分解

3.3　负载–时延关系与链路拥塞结构分析

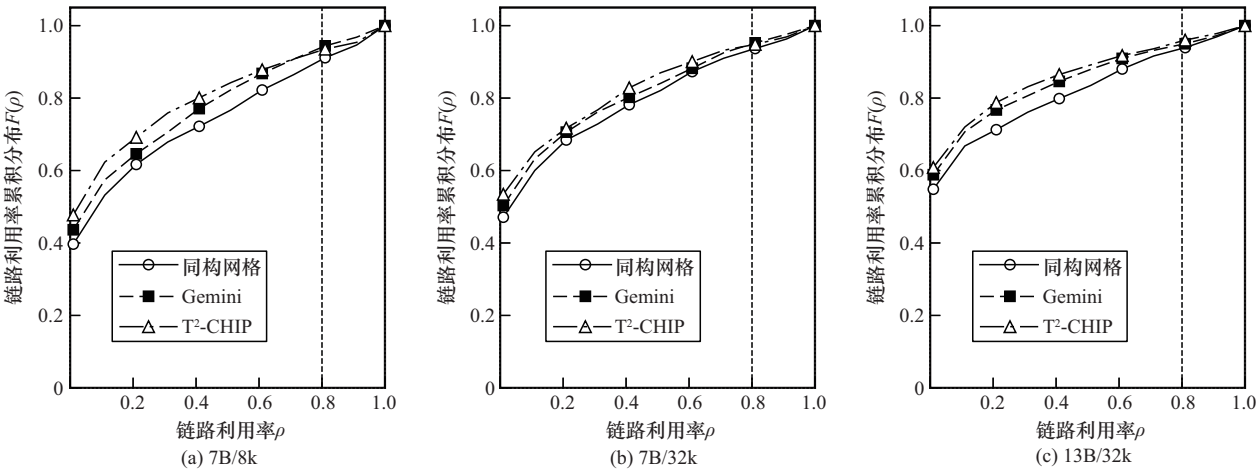
不同θ下的时延负载曲线如图7所示。从图7可以看出，与同构网格相比，Gemini和T²-CHIP的曲线整体向右移动且分离度增加，队列进入非线性阶段的拐点出现得更晚，稳定运行区间更大，这与关键截面获得更高的有效带宽、热点链路压力得到缓解的机制一致，也与SLO满足情况相互印证。

从解码阶段窗口化p99尾时延与θ的关系看，曲线右移意味着在给定p99阈值下的可承载负载提



(a) 7B/8k　(b) 7B/32k　(c) 13B/32k

图7　时延负载曲线



(a) 7B/8k　(b) 7B/32k　(c) 13B/32k

图 8　链路利用率累积分布曲线

高。T²-CHIP 与基线的曲线分离度随 θ 的增大而加大，表明容量配置提升了关键截面的有效带宽，降低了最忙链路利用率，从而推迟了队列进入非线性区的膝点，并扩大了 SLO 可行域，即同时满足 p99 与 $\rho_{\max}$ 约束的负载范围，这一趋势与链路侧 $\rho_{\max}$ 下降的趋势一致。

在链路侧，以链路利用率累积分布 $F(\rho)$ 与 Top-20 链路平均利用率两类统计量刻画负载结构。如图 8 所示，在相同链路利用率阈值 ρ 下，T²-CHIP 的 $F(\rho)$ 曲线整体更靠上，表明处于低至中等利用率区间的链路占比更高。与此同时，如图 9 所示，Top-20 链路平均利用率同步下降，热点显著扁平化，在目标负载点 $\rho_{\max}$ 被压制在 ρ_{tar}（0.8）以下，满足 SLO 中对最忙链路利用率的约束，网络处于更健康的拥塞状态。这些链路侧现象与网络 p99 的改进方向一致，指向容量配置对拥塞结构的直接优化作用。

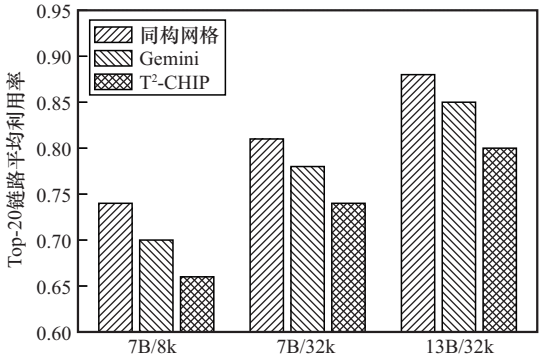


图9　链路热点统计

3.3.1　针对解码流量突发性的尾时延敏感性验证

为验证不同方案在解码阶段突发增强时的尾部稳定性，本文通过调节合成生成器的占空比 D 改变突发强度，保持平均字节率不变进行实验。如图 10(a)所示，随着突发因子 κ 增大，T²-CHIP 的尾部时延增长斜率较基线显著平缓。这归功于阶段 2 中引入的链路级等效突发因子 κ_e 对排队风险的放大

建模，以及阶段 3 在通道粒度上的闭环校正，确保了高突发条件下利用率的可控性。κ 描述生成器输入的 ON/OFF 时序突发结构，κ_e 为多流叠加与峰值窗口机制后的链路统计等价量，当其他条件固定时，增大 κ 通常会提升链路窗口序列的峰均比，从而导致 κ_e 随之增大。

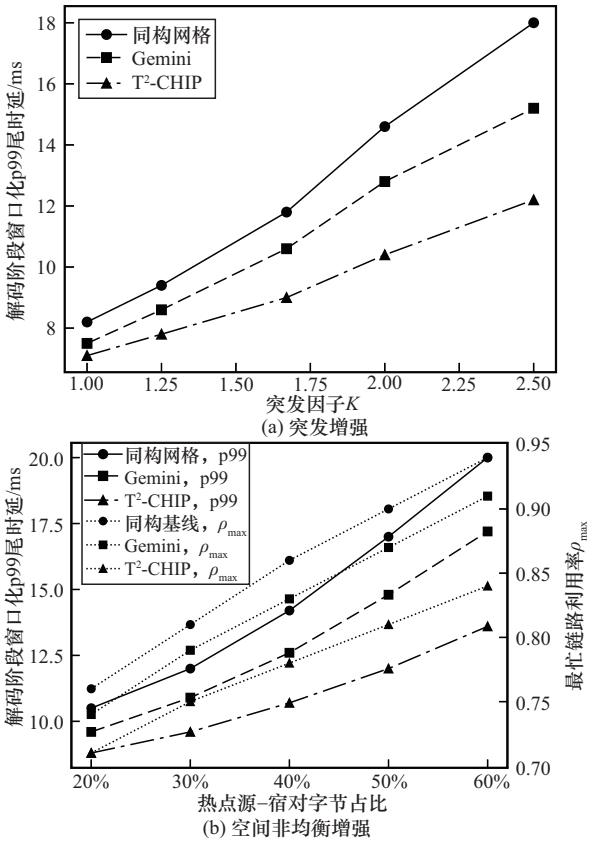


图 10　突发性与空间非均衡增强下的尾时延敏感性与拥塞结构变化

3.3.2　针对空间强非均衡性的热点扁平化能力验证

解码流量的强空间非均衡性使少数源–宿对与链路承载大部分通信量，导致热点链路的排队非线性放大。本文在保持总注入量与突发结构不变的条件下，通过增强热点集中度来考察其对系统尾部时延的影响。图 10(b)给出了不同热点占比下的窗口化 p99 尾时延以及对应的 $\rho_{\max}$ 变化趋势。实验结果显示，当热点更集中时，同构网格与 Gemini 的尾时延增长更快且 $\rho_{\max}$ 更早逼近拥塞阈值，说明其瓶颈截面对空间集中更敏感，T²-CHIP 的尾部抬升幅度更小，且 $\rho_{\max}$ 维持在更健康的区间。相比之下，T²-CHIP 在集中度增强时能够更好地控制 $\rho_{\max}$，保持更健康的拥塞结构。这是因为 T²-CHIP 通过供给侧的带宽重分配，确保容量定向到瓶颈截面，并通过闭环校正，维持更稳定的尾部性能。

3.4　针对瓶颈截面漂移的迁移验证与鲁棒性分析

解码阶段的跨芯粒通信热点会随模型规模、上下文长度与负载水平的变化而迁移，导致瓶颈截面在封装内网络中发生漂移，这意味着在单一服务形态下得到的静态容量配置可能不适用于其他服务形态。为此，本文提出基于热点集合重合率 η 的触发式更新规则：当在线统计得到的 η 低于预设阈值 $\eta_{\min}$ 时，触发离线重新画像（包括更新合成生成器参数），并在阶段 2/阶段 3 重新生成新配置。以下实验验证了这一策略的有效性。

3.4.1　热点漂移现象与量化刻画

3 种典型服务形态下 Top-10 热点链路的 λ_e^{p99} 热力图如图 11 所示，服务形态覆盖本文主要评估的 7B/8k、7B/32k 与 13B/32k 组合。结果表明，不同服务形态下热点链路的成员与强度分布发生变化，热点由一组链路迁移至另一组链路，瓶颈截面并非固定不变。

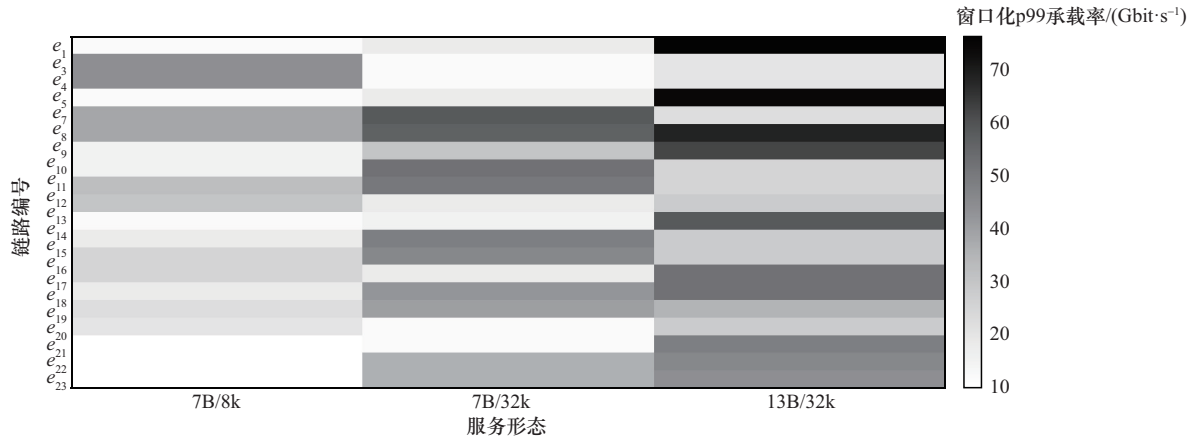


图 11　3 种典型服务形态下 Top-10 热点链路的 λ_e^{p99} 热力图

为量化热点集合的迁移程度，对于每一种服务形态，取窗口化p99承载率最高的前10条链路作为热点链路集合。为量化两种服务形态之间热点集合的一致性，将重合率定义为

$$\eta(i,j)=\frac{|\text{Top-10}(i)\cap\text{Top-10}(j)|}{10} \tag{11}$$

热点集合随着服务形态的变化而发生显著变化，具体数值如表6所示。

表6　热点集合重合率矩阵

服务形态	7B/8k	7B/32k	13B/32k
7B/8k	1.0	0.3	0.2
7B/32k	0.3	1.0	0.4
13B/32k	0.2	0.4	1.0

3.4.2　跨服务形态迁移实验设计

为验证瓶颈截面漂移对容量配置有效性的影响，本文设计跨服务形态迁移实验。服务形态A取7B/8k、并发32、θ=1.0，服务形态B取7B/32k、并发32、θ=1.0。由表6可得，两者热点集合重合率η = 0.3，低于本文设置的经验阈值$\eta_{\min}$ = 0.5，因此该迁移场景对应触发离线更新的典型情况。两种服务形态下NoP拓扑、路由策略与路由器微结构保持一致，总D2D带宽预算守恒。首先，在服务形态A下执行T²-CHIP的阶段1画像、阶段2容量配置与阶段3闭环校正，得到配置C_{A}。然后，将配置C_{A}直接迁移至服务形态B，并进行周期精确仿真。同时，在服务形态B下独立执行同样的流程得到更新配置C_{B}。

在服务形态B下对比3种配置C_0、C_{A}和C_{B}，其中，C_0为基线配置，采用同构网格均匀容量分配；C_{A}为直接迁移服务形态A下的容量配置；C_{B}为在服务形态B下重新画像并进行容量优化。采用解码阶段窗口化p99尾时延与最忙链路利用率$\rho_{\max}$作为主要指标。

3.4.3　跨服务形态迁移结果与分析

服务形态B下3种配置的解码阶段窗口化p99尾时延以及对应的$\rho_{\max}$对比结果如图12所示。与C_0相比，C_{A}在服务形态B下因配置失配导致尾部时延上升，$\rho_{\max}$增大，反映出瓶颈漂移削弱了静态配置的有效性；C_{B}通过重新画像和带宽再配置，显著降低了时延并控制了拥塞。

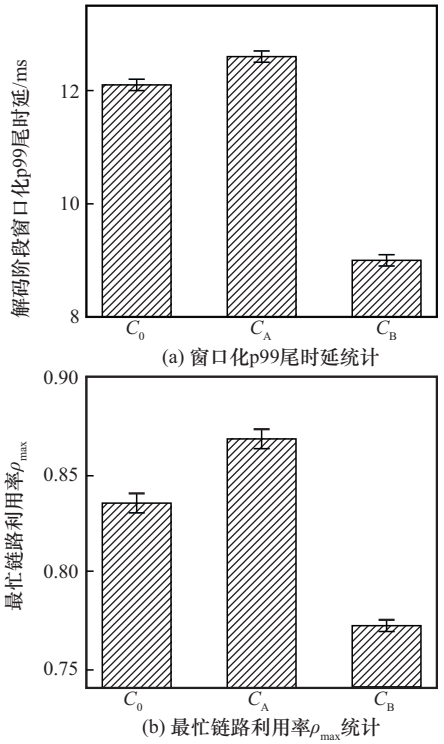


图12　服务形态B下不同配置的指标统计

本节验证了瓶颈截面漂移对容量配置有效性的影响，并说明了T²-CHIP通过重新画像与更新通道粒度容量配置能够适配漂移后的热点分布，从而在不同服务形态下稳定改善解码阶段窗口化尾部时延并抑制最忙链路拥塞。

3.5　消融与稳健性分析

消融实验统一固定了网格与路由器微架构、总TOPS、HBM容量与带宽以及总D2D预算，3种变体共享同一LTP、观测窗口W、负载因子θ与随机种子。容量配置使用相同的优化权重α与量化粒度，优化解经就近量化后在恒定预算下进行小步微调，每个负载点独立重复3次并基于窗口化统计计算指标。4种设置在不同模型规模/上下文组合下的解码阶段窗口化p99尾时延对比如图13所示。

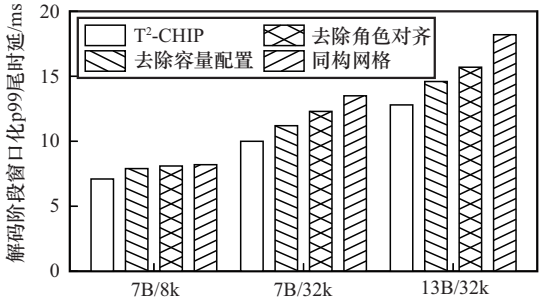


图13　p99消融实验对比

在相同消融设置下，最忙链路利用率$\rho_{\max}$与TTFT p99统计如图14所示。对比设置包括4类，T²-CHIP，角色对齐与容量配置同时启用；去除容量配置，仅角色对齐；去除角色对齐，仅容量配置；同构网格。

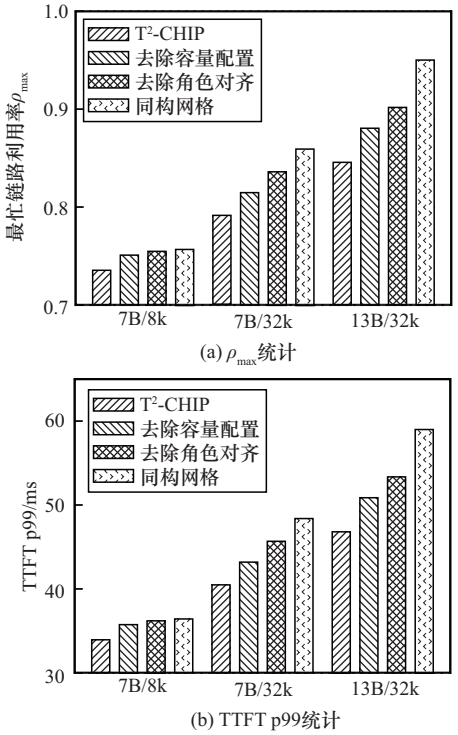


图14　$\rho_{\max}$和TTFT p99消融实验对比

消融结果表明，仅采用角色对齐或仅进行容量配置均能带来性能改进，但二者叠加才能同时取得更低的窗口化p99与更低的$\rho_{\max}$。图14(a)显示，容量配置在总预算不变的前提下直接降低$\rho_{\max}$，角色对齐更倾向于通过改变通信需求分布来缓解热点形成。二者叠加时$\rho_{\max}$最低。图14(b)显示，TTFT p99的变化与网络侧拥塞缓解一致，说明收益能够传导到端到端指标。

综上所述，随着模型规模增大、上下文变长与并发提高，各对比系统的窗口化p99普遍上升、TPS相应下降。在相同条件下，T²-CHIP能够稳定控制最忙链路利用率与尾时延的增加。当并发提升至32时，同构网格最早越过$\rho_{\max}=\rho_{\text{tar}}$拥塞警戒线，T²-CHIP保持$\rho_{\max}$不超过该阈值的负载区间更宽。总体而言，在长上下文、解码主导与中高并发场景下，T²-CHIP能够在固定预算内有效抑制链路拥塞、显著降低窗口化尾时延，并在吞吐量与时延之间保持稳定平衡。

LTP采用窗口化统计以刻画解码阶段的细粒度突发。窗口长度W会影响统计口径：较大的W可能稀释短时突发导致峰值压力被平滑，较小的W则可能放大随机抖动并降低统计稳定性。为考察W对LTP统计以及容量配置结论的影响，选取$W\in\{16,32,64,128\}$ ms。对于每个W，重新采集对应的LTP，并在阶段2、阶段3保持契约一致完成容量配置求解与闭环校正，结果如图15所示。随着W增大，窗口化p99的绝对数值整体呈下降或更平滑的趋势。本文关注相对改进与排序鲁棒性，因此用相同W进行比较仍有效。更重要的是，在不同W下，同构网格、Gemini与T²-CHIP的相对排序保持一致，且T²-CHIP相对同构网格的收益幅度变化较小，表明容量配置结论对窗口长度具有稳健性。综合统计稳定性与对token级突发的分辨能力，本文实验默认采用W=64 ms。

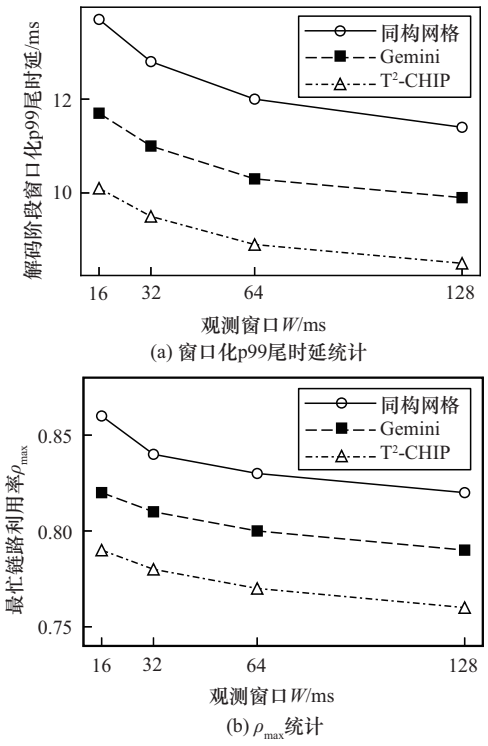


图15　不同观测窗口长度W下的性能对比

3.6　实现开销分析

T²-CHIP的优化自由度集中在封装内互连的容量空间配置，因此，其潜在实现代价主要体现在封装侧互连资源的分配偏置，而非芯粒重设计或互连规模扩张。

1) 动态功耗趋势

在固定拓扑、固定路由与固定路由器微结构的条件下，端到端通信量处于同一量级，因此SerDes/链路侧的传输能耗差异更接近由发送比特数决定的常数项。相比之下，拥塞导致的排队会显著增加路由器内部缓冲驻留与仲裁、交换相关的活动，从而增加与拥塞强相关的动态开销。为量化这一趋势，本文基于BookSim2的周期精确统计构造两个归一化指标。

归一化缓冲驻留量定义为

$$\widehat{O_{\text{buf}}}=\frac{\sum_{r}\sum_{v}\sum_{t}q_{r,v}(t)}{\sum_{r}\sum_{v}\sum_{t}q_{r,v}^{\text{hom}}(t)} \tag{9}$$

其中，$q_{r,v}(t)$表示周期t时路由器r的虚通道v中的驻留flit数，分母为同一工作负载下同构网格的对应统计量。归一化动态活动量代理定义为

$$\hat{A}=\frac{\sum_{s\in S}N_s}{\sum_{s\in S}N_s^{\text{hom}}},S=\{\text{ buf\_rd/wr,VA,SA,xbar... }\} \tag{10}$$

其中，N_s为BookSim2可直接统计的拥塞相关核心事件计数。由于方案间路由器微结构保持一致，在相同电压频率与工艺假设下，各类事件的单位能耗可视为常数。

如图16所示，相较同构网络，T²-CHIP在3种负载下的$\widehat{O_{\text{buf}}}$降至0.80、0.74、0.69，分别降低20%、26%、31%；相较Gemini，T²-CHIP分别进一步降低约11%、16%、20%。T²-CHIP的$\hat{A}$同样下降，分别为0.89、0.85、0.83。上述趋势说明，在总预算不变条件下，容量配置对热点截面的拥塞抑制不仅降低了尾时延，也同步降低了路由器内部排队与控制/交换活动强度，从而使动态功耗趋势下降。

2) 封装复杂度与实现成本

在本文约束设定下，NoP拓扑、路由与路由器/PHY微结构在不同方案间保持一致，SerDes通道粒度ΔCch与总通道预算（由总D2D带宽预算决定）也保持不变。因此，系统层面的新增成本并非源于总I/O数量、PHY宏单元数量或互连总规模的扩张。从系统总量上看，路由器与端口相关面积保持不变，总SerDes/PHY数量与对应宏单元面积保持不变，封装互连的总带宽规模与I/O资源占用保持不变，制造成本也不会随方案变化而线性增加。

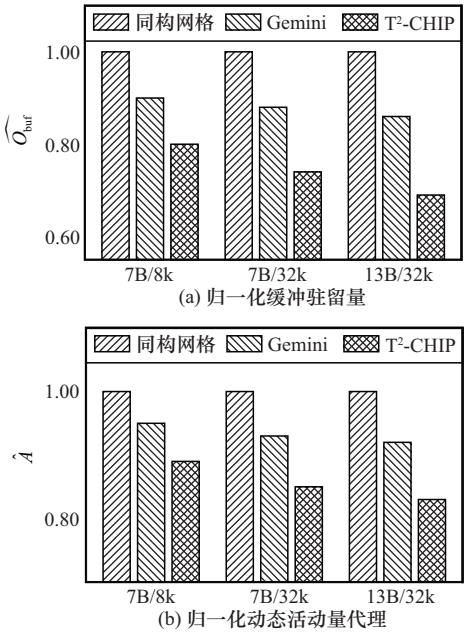


图16　不同工作负载下的缓冲驻留与路由器动态活动趋势

在上述总量不变的前提下，封装实现代价的潜在变化主要体现在局部通道资源的重新编排，即容量重分配可能使通道数在少数关键链路上出现一定程度的集中，从而提高热点区域的局部布线密度需求，并对走线拥塞、等长约束与信号完整性收敛带来更高压力，但该变化并不等价于新增端口或新增总通道。为将上述局部偏置显式量化为可复现的封装复杂度指标，本文以$n_e=\frac{C_e}{\Delta\text{Cch}}$组成的整体分布描述容量分配情况，并采用两项统计量刻画偏置程度：峰值密度倍率和离散度，计算式分别为

$$M_{\text{peak}}=\frac{\max\limits_{e}(n_e)}{\operatorname{mean}\limits_{e}(n_e)} \tag{11}$$

$$\text{CV}=\frac{\operatorname{std}\limits_{e}(n_e)}{\operatorname{mean}\limits_{e}(n_e)} \tag{12}$$

其中，M_{peak}反映最拥挤链路相对平均水平的放大倍数，CV反映整体分布的不均衡程度，二者越大，越可能对局部RDL/走线拥塞与SI收敛带来压力。在本文评估的3个负载点（7B/8k、7B/32k与13B/32k）上，T²-CHIP的M_{peak}分别为1.38、1.45、1.62，对应的CV分别为0.22、0.28、0.33。上述结果表明，容量重分配确实会提高少数关键链路的并行通道数，但偏置程度处于有限集中的范围内。在总通道预算不变的条件下，这种以局部密度换取关键截面带宽余量的配置更符合本文面向热点拥塞场

景的优化目标，同时避免将代价转化为新增封装层数、扩展总 I/O 或增配总通道等成本敏感因素。总体而言，T²-CHIP 在不扩展互连总规模与不增加总 SerDes 预算的前提下，通过可控的局部配置偏置获得热点截面的带宽余量与尾部稳定性，在封装实现方面具有较好的可落地性。

4　结束语

本文针对多芯粒 LLM 加速器封装内网络的性能瓶颈，提出推理协同优化方法 T²-CHIP。在固定拓扑、路由与统一路由器微结构，且总 D2D 带宽预算给定的平台复用约束下，T²-CHIP 通过任务角色对齐显式利用芯粒异构特性，并结合对解码阶段跨芯粒流量分布的分析，在总预算守恒条件下对逐链路带宽进行定向重分配，从而缓解热点链路排队并稳定控制尾部时延。仿真结果表明，在严格预算匹配与固定微架构条件下，相较同构网格，T²-CHIP 将解码阶段窗口化 p99 尾时延降低 25.8%，TTFT p99 降低 15.6%，TPS 提升 16.7%，同时满足最忙链路利用率约束。消融分析进一步表明，角色对齐与容量重分配具有互补性，协同作用能够更充分地提升关键截面有效带宽并抑制热点集中。

在实现代价方面，该方法在不扩展互连规模的前提下呈现动态功耗下降的趋势，并保持可控的封装配置复杂度与较低实现开销。未来工作将从以下方向展开：其一，探索将容量配置方法迁移至光互连或电光混合互连，评估距离无关带宽条件下的尾部性能边界；其二，与运行时调度策略耦合，形成“设计时优化–运行时适配”的闭环；其三，在制造缺陷、热约束与更严格离散化约束下开展规模化鲁棒性验证。总体而言，本文提出的尾部优先、预算守恒的带宽配置范式，为后摩尔时代多芯粒 LLM 推理系统的封装内网络设计提供了可复现且具备迁移性的参考路径。

参考文献：

[1] Jouppi N P, Hyun Yoon D, Ashcraft M, et al. Ten lessons from three generations shaped google’s TPUv4i: industrial product[C]//Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). Piscataway: IEEE Press, 2021: 1-14.

[2] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877-1901.

[3] Feng Y X, Xiang D, Ma K S. A scalable methodology for designing efficient interconnection network of chiplets[C]//Proceedings of the 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Piscataway: IEEE Press, 2023: 1059-1071.

[4] Li Y, Louri A, Karanth A. Scaling deep-learning inference with chiplet-based architecture and photonic interconnects[C]//Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC). Piscataway: IEEE Press, 2021: 931-936.

[5] Li C G, Jiang F, Chen S X, et al. Towards scalable GPU system with silicon photonic chiplet[C]//Proceedings of the 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE). Piscataway: IEEE Press, 2024: 1-6.

[6] Cai J W, Wu Z T, Peng S, et al. Gemini: mapping and architecture co-exploration for large-scale DNN chiplet accelerators[C]//Proceedings of the 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Piscataway: IEEE Press, 2024: 156-171.

[7] Yu Z K, Liang S W, Ma T Y, et al. Cambricon-LLM: a chiplet-based hybrid architecture for on-device inference of 70B LLM[C]//Proceedings of the 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO). Piscataway: IEEE Press, 2024: 1474-1488.

[8] Dao T, Fu D Y, Ermon S, et al. Flashattention: fast and memory-efficient exact attention with io-awareness[J]. Advances in Neural Information Processing Systems, 2022, 35: 16344-16359.

[9] Zhong Y, Liu S, Chen J, et al. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving[C]//Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). Berkeley: USENIX Association, 2024: 193-210.

[10] Kwon W, Li Z H, Zhuang S Y, et al. Efficient memory management for large language model serving with PagedAttention[C]//Proceedings of the 29th Symposium on Operating Systems Principles. New York: ACM Press, 2023: 611-626.

[11] Zhou M X, Xu W H, Kang J, et al. TransPIM: a memory-based acceleration via software-hardware co-design for transformer[C]//Proceedings of the 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Piscataway: IEEE Press, 2022: 1071-1085.

[12] Ding Y, Liu C B, Duan M X, et al. HAIMA: a hybrid SRAM and DRAM accelerator-in-memory architecture for transformer[C]//Proceedings of the 2023 60th ACM/IEEE Design Automation Conference (DAC). Piscataway: IEEE Press, 2023: 1-6.

[13] Wang M D, Wang Y, Liu C, et al. Network-on-interposer design for agile neural-network processor chip customization[C]//Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC). Piscataway: IEEE Press, 2021: 49-54.

[14] Loh G H, Swaminathan R. The next era for chiplet innovation[C]//Proceedings of the 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE). Piscataway: IEEE Press, 2023: 1-6.

[15] Huang P K, Lu C Y, Wei W H, et al. Wafer level system integration of the fifth generation CoWoS®-S with high performance Si interposer at 2500 mm2[C]//Proceedings of the 2021 IEEE 71st Electronic Components and Technology Conference (ECTC). Piscataway: IEEE Press, 2021: 101-104.

[16] 李沛杰, 沈剑良, 郭威, 等. 面向软件定义晶上系统的安全互连接口架构[J]. 通信学报, 2024, 45(10): 41-54.

Li P J, Shen J L, Guo W, et al. Secure interface architecture for the software defined system on wafer[J]. Journal on Communications, 2024, 45(10): 41-54.

[17] Wang X H, Wang Y F, Jiang Y T, et al. On task mapping in multi-chiplet based many-core systems to optimize inter- and intra-chiplet communications[J]. IEEE Transactions on Computers, 2025, 74(2): 510-525.

[18] Duan Y Y, Liu X C, Yu Z P, et al. RLPlanner: reinforcement learning based floorplanning for chiplets with fast thermal analysis[C]//Proceedings of the 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE). Piscataway: IEEE Press, 2024: 1-2.

[19] Mallya N B, Strikos P, Goel B, et al. A performance analysis of chiplet-based systems[C]//Proceedings of the 2025 Design, Automation & Test in Europe Conference (DATE). Piscataway: IEEE Press, 2025: 1-7.

[20] Yang H J, Fang J, Hou Y M, et al. Reinforcement learning-driven adaptive prefetch aggressiveness control for enhanced performance in parallel system architectures[J]. IEEE Transactions on Parallel and Distributed Systems, 2025, 36(5): 977-993.

[21] Zhang J M, Wang X Y, Ye Y Y, et al. M2M: a fine-grained mapping framework to accelerate multiple DNNs on a multi-chiplet architecture[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2024, 32(10): 1864-1877.

[22] Zhang J M, Fan X, Ye Y Y, et al. INDM: chiplet-based interconnect network and dataflow mapping for DNN accelerators[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, 43(4): 1107-1120.

[23] Das A, Russo E, Palesi M. Multi-objective hardware-mapping co-optimisation for multi-DNN workloads on chiplet-based accelerators[J]. IEEE Transactions on Computers, 2024, 73(8): 1883-1898.

[24] Odema M, Chen L K, Kwon H, et al. SCAR: scheduling multi-model AI workloads on heterogeneous multi-chiplet module accelerators[C]//Proceedings of the 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO). Piscataway: IEEE Press, 2024: 565-579.

[25] Mishty K, Sadi M. Chiplet-gym: optimizing chiplet-based AI accelerator design with reinforcement learning[J]. IEEE Transactions on Computers, 2025, 74(1): 43-56.

[26] Nalla P S, Haque E, Liu Y T, et al. CLAIRE: composable chiplet libraries for AI inference[C]//Proceedings of the 2025 Design, Automation & Test in Europe Conference (DATE). Piscataway: IEEE Press, 2025: 1-7.

[27] 李雯, 王颖, 何银涛, 等. SMCA: 基于芯粒集成的存算一体加速器扩展框架[J]. 电子与信息学报, 2024, 46(11): 4081-4091.
Li W, Wang Y, He Y T, et al. SMCA: a framework for scaling chiplet-based computing-in-memory accelerators[J]. Journal of Electronics & Information Technology, 2024, 46(11): 4081-4091.

[28] Du S T, Zheng L Q, Parvathy A M, et al. 3D-CIMlet: a chiplet co-design framework for heterogeneous in-memory acceleration of edge LLM inference and continual learning[C]//Proceedings of the 2025 62nd ACM/IEEE Design Automation Conference (DAC). Piscataway: IEEE Press, 2025: 1-7.

[作者简介]

方娟（1973– ），女，辽宁鞍山人，博士，北京工业大学教授、博士生导师，主要研究方向为高性能计算、智能计算、边缘智能、云边协同、推理优化等。

潘晨阳（2000– ），男，山西运城人，北京工业大学硕士生，主要研究方向为高性能计算、芯粒互连。

古明辉（1999– ），男，河南开封人，北京工业大学硕士生，主要研究方向为高性能计算、芯粒互连。

李硕朋（1989– ），男，辽宁抚顺人，博士，北京工业大学助理研究员、硕士生导师，主要研究方向为软件定义网络、网络功能虚拟化、网络可靠性传输、空天地一体化网络、物联网、边缘计算、深度强化学习等。

陈慧杰（1989– ），男，河南新乡人，博士，北京工业大学讲师、硕士生导师，主要研究方向为智能物联网、泛在感知、边缘计算、物联网系统安全与隐私保护等。

翟冉（1998– ），女，河北唐山人，北京工业大学博士生，主要研究方向为高性能计算、片上网络。