CAICT 中国信通院

中国人工智能产业发展联盟
Artificial Intelligence Industry Alliance

集智 专题报告

大模型推理优化关键技术及应用实践研究报告

（2026 年）

中国信息通信研究院人工智能研究所
中国人工智能产业发展联盟

2026年3月

版权声明

前　言

大模型推理作为人工智能技术从实验室走向产业应用的“最后一公里”，承载着将模型能力转化为实际业务价值、平衡服务质量与成本投入的核心使命。随着生成式 AI、智能体（Agent）、多模态交互等技术的爆发式发展，推理需求呈现指数级增长。行业数据显示，2025 年全球大模型推理计算量较上年提升 100 倍以上，同时，推理预算也在持续攀升，成为企业规模化落地的关键瓶颈。与此同时，不同场景对推理服务的差异化诉求（如低时延、高并发、长上下文处理）日益凸显，传统单点优化技术已难以应对“效果-性能-成本”的多目标协同，亟需构建全链路、系统性的推理优化体系。

本报告立足产业实践与技术演进，系统梳理大模型推理优化的技术路径与落地脉络。首先，剖析推理优化催生背景与概念特性；梳理当前围绕多样化场景适配、算力成本平衡、模型特性适配的核心挑战，揭示产业落地痛点。然后，根据关键技术发展，拆解模型、引擎、系统三级优化体系的核心方法与适配逻辑；结合产业生态演进趋势，分析从单点优化到“模型-架构-场景”协同优化的发展方向。再次，通过金融、运营商、电力、农业等行业案例验证技术落地价值。最后，提出技术与产业展望与建议。

本报告力求为大模型推理优化领域的技术选型、方案设计与产业落地提供参考，因行业发展迅速，内容难免存在疏漏，恳请各位专家与读者批评指正。

目　　录

图 目 录

表 目 录

一、大模型推理优化概况

大模型推理平台是指为千亿级参数模型提供高效推理服务的工程化基座，涵盖轻量化工具（如推理引擎、工具包）与集成化系统（如公有云服务、私有化部署系统、混合部署系统、边缘计算系统），一般通过“硬件-软件-模型-服务”的协同优化，实现高精度、低延迟、高并发与低成本的规模化服务输出，支持全场景 AI 服务化落地。随着大模型技术的飞速发展和企业智能化转型需求的不断攀升，大模型落地应用关注焦点正从训练环节转向推理环节。在此过程中，行业需求已从构建功能全面、用户友好且灵活的推理平台，逐步深化到解决实际落地中由“**效果-性能-成本**”构成的多目标协同难题。推理优化技术作为破解该难题的核心抓手，其重要价值正在大模型规模化应用中愈发凸显。

（一） 大模型推理成为新的落地焦点

大模型产业正迈向规模化落地的关键转型期，其落地焦点正从训练走向推理。作为连接技术创新与产业应用的核心枢纽，大模型推理正推动技术创新加速从验证迈向规模化应用。这一变革由市场需求、算存供给、成本经济与应用场景等多重因素共同驱动，标志着大模型正式进入规模化商业落地与价值兑现的新阶段。

需求侧，大模型服务调用量与推理计算量呈现爆发式增长。推理服务调用量暴增，OpenAI 2025 年 12 月发布的《2025 年企业人工智能状况报告》显示，过去 12 个月，ChatGPT 企业版 API 推理 Token 消耗暴增 320 倍，企业端消息量增长 8 倍。**推理计算量翻倍，**黄仁勋

在 2025 年英伟达 GTC 大会上表示，由于代理型 AI（Agentic AI）和推理能力的发展，目前所需的计算量轻松达到了去年预估的 100 倍。**服务平均序列长度攀增，**从 2023 年最大序列长度 4K 增长到当前达 128K，两年间增长至 32 倍，体现了当前大模型推理服务的任务复杂性与交互深度性。**供给侧，算存资源、成本投入等配置重心正在向推理倾斜。算力供给层面，**全球推理算力持续增长，2026 年计算工作负载中推理占比将提升至 66%；我国市场亦增速迅猛，2026 年推理算力市场规模将达 876.5 亿元，较 25 年的 438.3 亿元接近翻倍[1]。**存储供给层面，**推理应用对长记忆数据存储的需求显著提升，2025 年，DRAM/SSD(闪存存储)/HDD(机械硬盘存储)价格指数累计增长 327.59%/166.28%/66%。**业务预算层面，**2024 年 OpenAI 推理业务预算达 23 亿美元，为训练 GPT-4 的 1.5 亿美元的 15 倍。可见大模型推理不同于训练的一次性投入，其伴随部署服务的持续性消耗，进一步导致推理账单的占比增长[2]。

（二） 大模型推理优化的概念与目标

1.定义与范围

大模型推理优化是指在保障模型服务等级目标（SLO）的前提下，通过一系列覆盖模型、引擎、系统（软/硬件）及服务全链路的技术手段与工程实践，系统性提升推理性能、降低运营成本的过程。其核心目标在于兼顾“效果-性能-成本”的协同优化，实现三者之间的动态平

[1] 弗若斯特沙利文, 中国推理算力市场追踪报告 2025H1
https://www.frostchina.com/content/insight/detail/694f9b974a7a7390decf1c08
[2] A.I News Hub, AI Inference Costs 2025: Why Google TPUs Beat Nvidia GPUs by 4x
https://www.ainewshub.org/post/ai-inference-costs-tpu-vs-gpu-2025

衡与最优权衡，从而支撑大模型技术规模化、可持续化的商业落地。

从 AI 全生命周期看，训练阶段的核心任务是通过海量数据学习模型参数，追求的是模型能力的上限，其过程一般具备离线、长周期、计算密集的特点。而推理阶段则是将已训练好的模型部署为在线或离线服务。在线服务一般要求实时响应用户请求，其核心诉求是效率、稳定与经济性；离线服务一般要求大批量生成结果，其核心诉求是吞吐、稳定与经济性。

从推理内部环节看，大模型推理优化贯穿于“压缩-部署-推理-服务”四个环节。压缩环节关注如何在可接受的精度损失范围内，通过量化、剪枝、蒸馏等技术减小模型体积与计算复杂度；**部署环节**关注如何高效地将模型加载至目标硬件环境，涉及容器化、冷启动加速，以及初始的网络/存储/计算资源配置等；**推理环节**是核心执行阶段，涵盖批处理调度、显存管理、高性能算子等引擎级优化，以及分布式推理、推理架构设计等系统级优化；**服务环节**则面向最终用户体验，包括 API 能力、请求调度、资源调度、负载均衡、弹性扩缩容、监控告警等能力。

从面向对象来看，推理优化的实施对象呈现出多层次、多样化的特征。在工具层面，以轻量化推理引擎与压缩工具为代表，可提供高性能的底层执行能力，迅速集成新兴技术并实现产业赋能。**在系统层面，**涵盖轻量化按需订阅模式的模型即服务（MaaS），自主可控定制化模式的私有化部署平台，本地化即插即用模式推理一体机，分布式实时响应模式的云-边-端协同推理系统，弹性伸缩平衡模式的混合部

署系统等多种类型。这些系统不仅集成了推理引擎能力，还提供了完整的模型管理、资源调度、安全管控与可观测性体系，满足不同规模企业的需求。

2.主要发展阶段

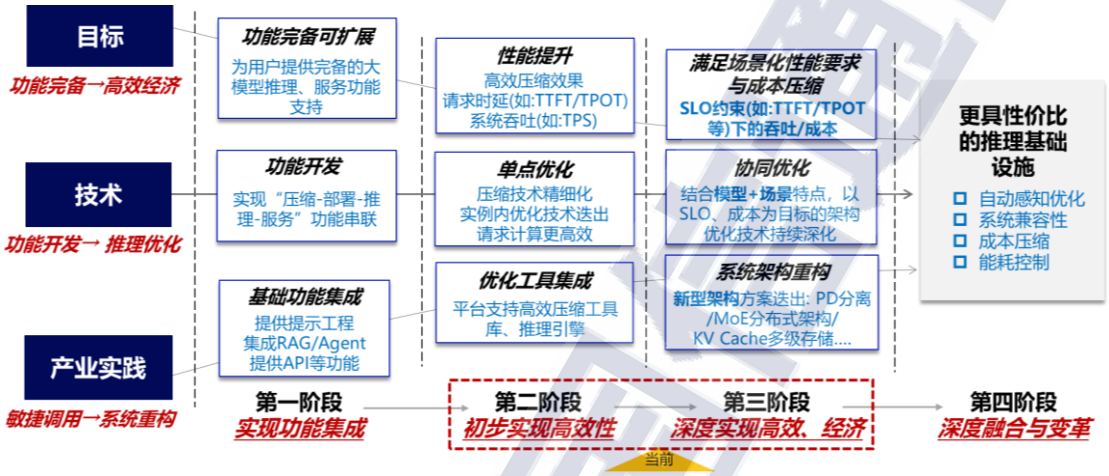


图 1 大模型推理基础设施发展主要阶段

大模型推理基础设施已完成初期功能集成，正走向系统级推理优化的深水区，各阶段围绕技术能力升级与产业价值落地，形成递进式目标。

第一阶段为功能集成阶段，核心目标是实现推理服务的功能完备与可扩展。该阶段聚焦基础能力搭建，通过集成模型管理、部署调度、接口服务、运行监控等功能，打通从“训练输出模型”到“推理提供服务”的全流程链路。实践中，平台重点支持 RAG、Agent 等扩展能力，并提供标准化 API 接口，确保大模型推理服务能适配多样化业务接入需求，完成从“技术可用”到“服务可用”的跨越。

第二阶段为初步性能提效阶段，核心目标是实现服务局部性能提升与推理单点优化。在功能完备的基础上，该阶段通过压缩技术精细

化，以及并行策略、显存优化、计算优化、批调度优化等实例内推理优化技术演进，实现服务请求计算效率的提升；同时，平台集成压缩工具库、推理引擎等工具，以降低请求时延（TTFT/TPOT/端到端时延）、提升系统吞吐（TPS/QPS）为核心指标，实现推理性能的单点突破。此阶段的典型特征是技术聚焦于局部效率提升，通过工程化手段初步释放算力效能，为规模化应用提供性能支撑。

第三阶段为深化提效与经济落地阶段，核心目标是实现场景协同优化与系统架构重构。该阶段不再局限于单点的技术优化与性能提升，而是结合模型特性与场景需求，以 SLO 为导向，推动“模型-架构-场景”的协同优化。实践中，预填充-解码（PD）分离、KV Cache 多级存储、混合专家（MoE）分布式架构、注意力-前向反馈（AF）分离等新型系统架构成为主流，在满足场景化性能要求的同时，显著降低推理成本，实现高效与经济的双重目标。当前大模型推理基础设施正处于第二、第三阶段。

第四阶段为深度融合与变革阶段，核心目标是构建更具性价比与自适应性的推理基础设施。该阶段通过自适应感知、系统级兼容性设计、全链路成本压缩等方式，推动推理服务与业务场景深度融合，形成具备自优化、高兼容、低能耗的新一代推理体系。

3.核心目标

在此背景下，大模型推理的落地目标经历了从局部性能指标提升到多目标协同平衡的演进。

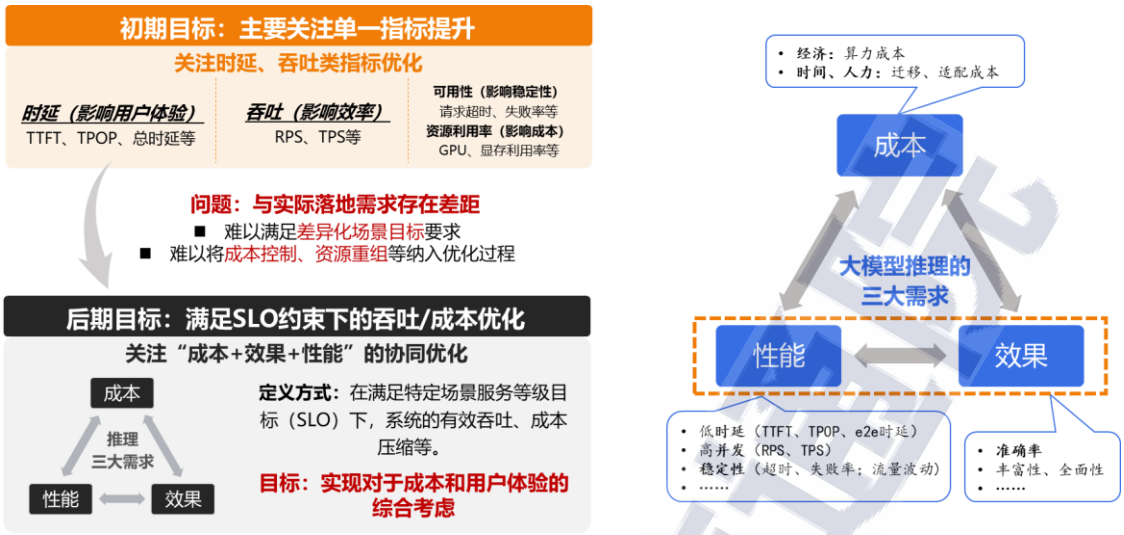


图 2 大模型推理核心目标

初期聚焦单一指标的性能提升。该阶段以技术验证为核心导向，主要关注时延、吞吐等关键指标的局部优化。然而，这种单一维度的优化与产业落地需求存在明显差距：**一方面，**难以适配多样化场景的差异化性能要求，例如，高并发场景对吞吐的需求与长上下文场景对时延的诉求难以兼顾；**另一方面，**未将成本控制、资源重组等实践因素纳入优化过程，导致技术方案的规模化落地能力受限。

当前目标为面向 SLO 约束的多目标协同优化。随着大模型进入规模化商业落地阶段，推理优化的目标转向“模型-架构-场景”的协同平衡，**即在满足特定场景 SLO 的前提下，实现系统有效吞吐、成本压缩与效果保障的综合最优。**其中，**效果**主要体现在特定业务场景下生成结果的可用性、相关性、准确性与安全性；**性能**主要体现在时延、吞吐量、服务稳定性（如失败率、P99 延迟）等可量化的服务质量（QoS）指标上；**成本**涵盖算力成本（GPU/NPU 小时数、显存占用），以及人力与时间成本（如模型迁移、适配、运维的复杂度）。

二、大模型推理的主要挑战

（一）多样化场景的适配

大模型技术的广泛应用，催生了极其丰富的落地场景。然而，不同场景对推理服务的核心诉求存在显著差异，形成了以低时延、高并发、流量波动与长上下文为代表的四大典型场景。**一是低时延场景，**如智能客服、实时对话系统，要求服务在用户可感知的时间内（通常为数百毫秒内）返回首个 Token（Time to First Token, TTFT）；**二是高并发场景，**如批量内容生成、大规模数据标注、离线报告撰写等，追求高吞吐量（RPS/TPS），以最大化单位时间内的任务处理能力；**三是长上下文场景，**如基于 RAG 的知识问答、多轮 Agent 协作，随着上下文窗口从数千 Token 扩展至百万级别，KV Cache 的显存占用呈线性甚至超线性增长，迅速耗尽 GPU 高带宽内存（HBM），成为长上下文场景下的主要瓶颈。**此外，流量波动是几乎所有在线服务都面临的现实挑战。**部分场景的业务流量呈现周期性（如工作日高峰、节假日低谷）或突发性（如营销活动、热点事件）特点。静态推理系统在闲时会造成资源浪费，在忙时则会因过载而拒绝服务或延迟飙升，因此，系统的流量自适应与弹性能力尤为关键。

（二）高质量算力需求与成本控制的平衡

高质量算力需求与成本控制的平衡是大模型规模化落地的另一挑战。一方面，复杂场景对算力的性能、稳定性提出严苛要求，需依托高性能硬件与优化技术保障服务质量；**另一方面，**推理阶段持续的算力消耗已成为企业核心成本负担，如何高效有机复用起存量算力、

适配异构算力资源、实现跨场景协同调度，成为破解这一矛盾的关键。**前序算力资源的整合、复用面临多重阻碍。**企业存量的早期 GPU 等资源，因硬件架构、软件生态与大模型推理需求可能存在不兼容问题，直接部署易出现性能短板。**跨场景算力协同调度面临多重困境。**异构算力的广泛应用，要求调度系统实现资源动态按需分配，但调度决策、资源隔离、动态适配三大挑战凸显。以上挑战不仅涉及硬件选型与软件适配问题，更考验企业资源整合与架构设计的系统性能力。

（三）模型特性与发展需求的适配

大模型技术正处于高速迭代期，其架构与能力的演进日新月异。从传统的 Dense 架构向 MoE 架构、从单一语言向原生多模态、从数千 Token 上下文向百万级长序列持续跃迁。大模型的飞速发展对底层推理基础设施亦提出了严峻挑战：工程化方案必须具备高度的前瞻性与灵活性，能够快速适配新型模型的计算与存储特性，避免成为性能瓶颈，制约模型能力的充分释放。**模型的快速演进推动推理优化从“适配模型”迈向“协同演进”的新阶段。**未来的大模型推理平台需提供对 MoE、多模态、超长序列等前沿特性的原生支持能力，构建从硬件存储到软件调度的全栈优化体系。

三、大模型推理优化关键技术

（一）模型层面

模型层是大模型推理优化体系的源头，该层聚焦如何让模型本身更轻、更快、更易部署，通过结构升级、参数压缩、算子重构与算法创新，从根本上削减推理阶段的计算与存储开销。**相较于引擎层聚焦**

执行效率，模型层关注源头优化，通过结构重构与参数压缩，从根本上减少推理阶段的计算与存储开销。主要方向包括模型压缩、MoE 稀疏化架构以及算法创新。为引擎层和系统层提供了更轻量、更高效的模型形态，为后续引擎执行与系统调度奠定了“算少、算快、算得稳”的结构基础。

1.模型压缩

模型压缩技术是解决大模型在资源受限环境中部署难题的关键途径，其核心目标是在尽可能保持模型性能的前提下，显著降低模型的存储占用和计算需求。通用压缩技术主要包括量化、知识蒸馏、剪枝和稀疏化等方法。

通过剪枝、量化、蒸馏、低秩分解等通用压缩技术，可有效减少模型参数量，降低计算复杂度。量化通过降低权重和激活的数值精度减少存储开销,剪枝通过去除冗余连接降低模型密度，蒸馏则通过“教师-学生”机制保留模型的关键知识，低秩分解进一步在矩阵层面削减计算需求。这些方法的共性目标，是在最小化精度损失的同时，实现显存占用与计算时延的显著下降[3]。

大模型推理场景下更强调“高比例压缩-低精度损失-高性能推理”的均衡。研究表明，适度的量化与结构化剪枝可以在精度下降不足 1%的前提下，使推理速度提升 30%—50%，显著改善服务响应时延。同时，低秩分解与混合精度推理技术正在成为主流方向，通过矩阵分解

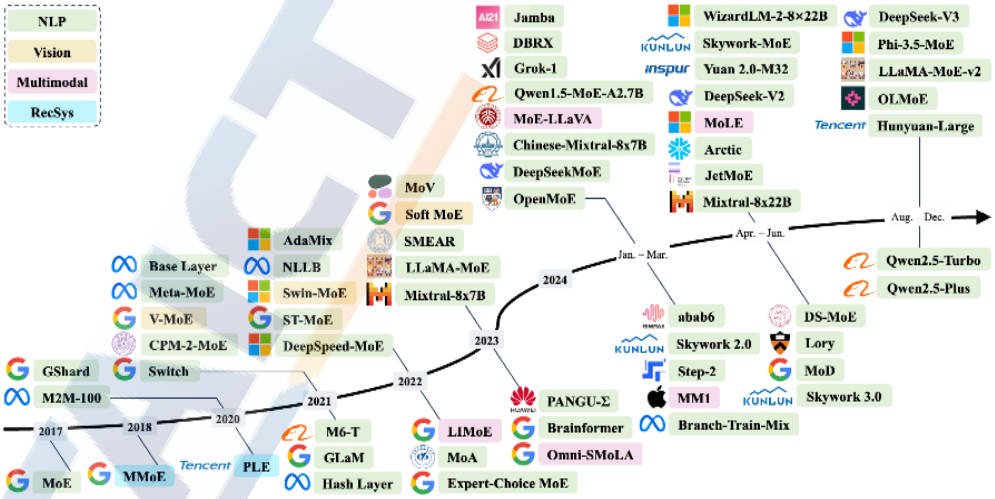
[3] Zhu, X. et al. (2023). A Survey on Model Compression for Large Language Models. arXiv:2308.07633. https://arxiv.org/abs/2308.07633

与计算精度分层，实现算子级别的高效计算[4]。

大模型压缩技术逐步向“无重训练压缩”与“自适应压缩”演进。传统压缩往往依赖重新训练以恢复精度，但随着模型规模扩大，该代价愈发不可接受。近年来出现的自动压缩与敏感度感知量化方法，可在不额外训练的情况下，自动识别重要参数并进行动态压缩，从而降低部署复杂度，提升推理阶段的可迁移性与可扩展性。

2.MoE 架构

随着参数规模指数级增长，稠密计算已难以兼顾性能与能效，混合专家（Mixture of Experts, MoE）模型架构以“按需激活”的稀疏计算模式，为推理优化提供了新的思路。其核心思想可追溯至集成学习和传统深度学习时代，经历了从早期方法到大语言模型应用、多模态模型应用的完整演进过程[5]，在当今大模型时代迎来了高速发展和广泛普及。

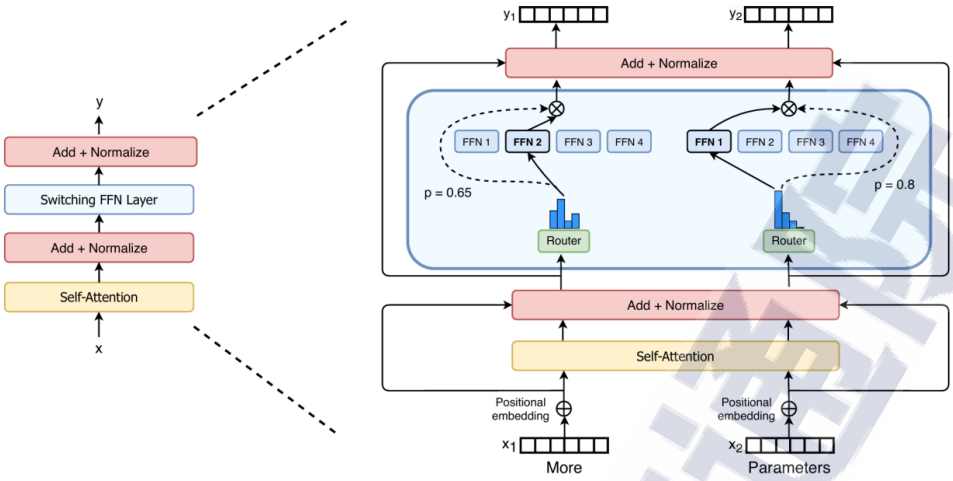


来源：arXiv:2407.06204v3

图 3 近年典型 MoE 模型发布时间

[4] Wang, W. et al. (2024). Model Compression and Efficient Inference for Large Language Models: A Survey. arXiv:2402.09748. https://arxiv.org/abs/2402.09748

[5] Weilin Cai, et al. "A Survey on Mixture of Experts in Large Language Models" (TKDE), 2025. https://arxiv.org/pdf/2407.06204

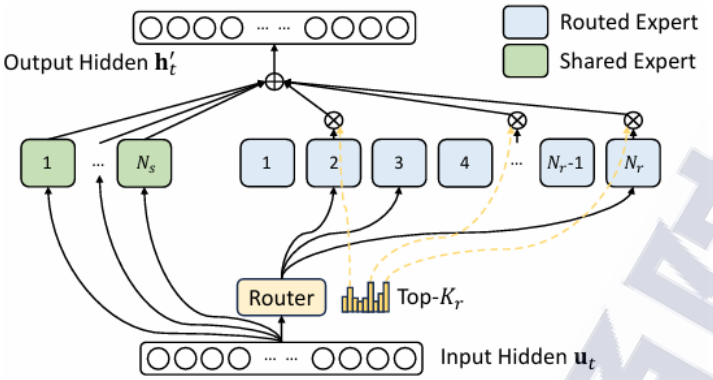


来源：arXiv:2101.03961v3

图 4 MoE 模型架构示意图

MoE 架构通过在推理阶段仅激活少量“专家网络”，可显著降低计算负载。其核心机制在于门控网络（Gating Network）根据输入特征动态选择专家，使每次推理仅涉及少量参数计算，从而实现计算稀疏化与显存占用优化。这种结构使得模型整体参数规模可扩展至万亿级，而单次推理计算量保持在可控范围[6]。MoE 架构在带来效率优势的同时，也引入新的工程挑战。由于推理阶段需完成专家选择、激活分配与通信调度等核心步骤，额外引入了路由开销与负载不均问题。针对这一挑战，多种负载均衡策略与专家选择算法不断提出，如基于梯度统计或样本重要性的门控机制，可在保证推理稳定性的同时提升专家利用率。

[6] Yang, Z. et al. (2024). A Closer Look into Mixture-of-Experts in Large Language Models. arXiv:2406.18219. https://arxiv.org/abs/2406.18219



来源：arXiv:2405.04434v5

图 5 DeepSeekMoE 模型架构示意图

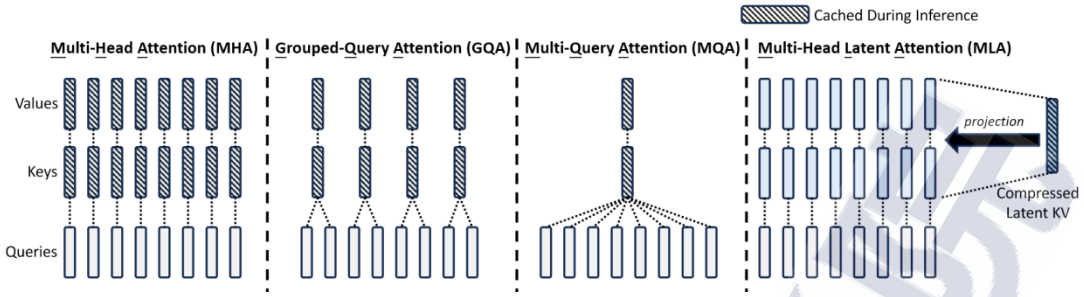
专家细粒度分割与动态负载均衡是 MoE 模型的主要优化趋势。传统 MoE 架构通常采用固定数量和规模的专家模块，专家粒度较粗，容易导致路由决策刚性、负载分配不均等问题。随着模型规模的持续扩大与推理任务多样化，细粒度专家设计成为新的优化方向。**一方面，**专家细粒度分割通过将原有大规模专家进一步拆分为更小的子专家单元，使模型能够以更高的分辨率匹配不同输入模式，显著提升专家利用率与激活灵活性。以 DeepSeekMoE 代表的研究提出，将专家划分为多层子专家结构，通过共享专家参数与跨层专家融合，实现“多任务共享+精准激活”的稀疏计算机制，从而在保持整体模型规模不变的前提下，进一步降低单次推理的显存占用与计算成本[7]。**另一方面，**在推理阶段通过监测专家激活分布与实时算力利用率，动态调整门控策略与专家调度策略，避免高频专家过载以及低频专家闲置引发资源浪费。研究表明，引入自适应门控与分层负载反馈后，模型整体吞吐可提升 2–3 倍，且可显著降低通信瓶颈[8]。

[7] DeepSeek-AI Research Team. (2024). DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. arXiv:2405.04434. https://arxiv.org/pdf/2405.04434
[8] DeepSeek-AI Research Team. (2024). DeepSeek-V3 Technical Report. arXiv:2412.19437. https://arxiv.org/pdf/2412.19437

3.算法优化

注意力机制改造与解码并行加速是当前模型提效热点。一方面，高效注意力机制持续改造。大模型推理多为自回归生成，Transformer 的 KV Cache 成为解码（Decode）阶段显存消耗的主要瓶颈。产学界围绕这一问题陆续提出多种注意力改进方案：多查询注意力（Multi-QueryAttention, MQA）通过共享一套 KV 实现缓存压缩，分组查询注意力（Grouped-QueryAttention, GQA）在性能与显存间进行折中。DeepSeek-V2 提出多头潜在注意力（Multi-head Latent Attention, MLA），进一步引入低秩压缩机制，在保留各头独立权重的同时显著压缩 KV Cache，实现显存压缩与模型性能提升的兼顾。**另一方面，**通过将 Decode 阶段并行化实现加速的算法已得到普遍落地。投机采样（Speculative Decoding）关注生成–验证过程的并行路径。通过引入一个草稿生成阶段和一个目标模型验证阶段，将推理中多个 Token 的生成与验证并行执行，从而打破了纯自回归的串行瓶颈。其在小批量场景下可将延迟降低约 2-3 倍[9]。DeepSeek 提出的多 Token 预测（Multi-Token Prediction, MTP）关注生成本身的并行化。通过让模型在每次前向过程中预测多个未来 Token，从模型内部优化设计出发，削减自回归推理中每个 Token 生成的延迟。

[9] Zhao, J. et al. (2023). Fast Inference from Transformers via Speculative Decoding. arXiv:2211.17192. https://arxiv.org/abs/2211.17192



来源：arXiv:2405.04434v5

图 6 MHA, MQA, GQA, MLA 架构图

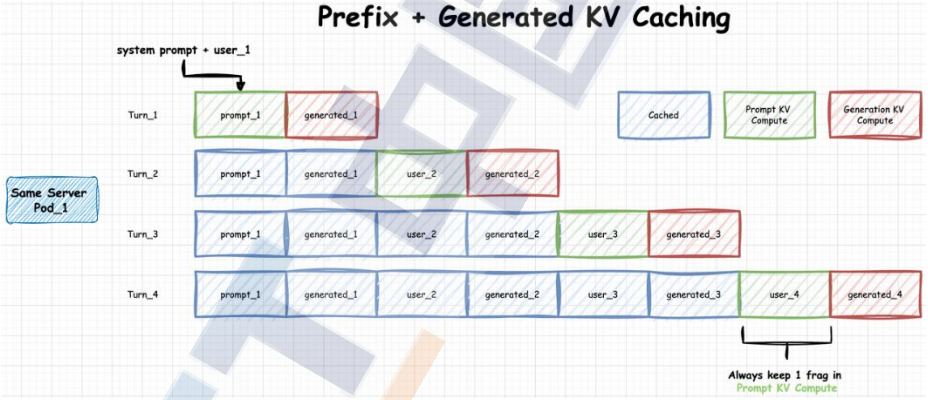
（二）引擎层面

引擎层是大模型推理体系中的关键枢纽，承担着将模型能力转化为可被系统高效执行、可直接支撑服务化部署的核心功能。作为模型与系统之间的执行层，其将训练好的模型转化为可高效推理的运行时实例，聚焦如何高效计算。**不同于系统架构层对跨节点资源调度与服务拓扑的全局设计，其优化边界聚焦于单实例或轻量集群内的执行效率，为上层系统架构提供高性能、低延迟的基础计算能力支撑。**当前的核心技术围绕显存管理、计算优化、并行加速、调度策略等关键环节展开，旨在解决 I/O 瓶颈、计算效率、资源利用等挑战。

1.显存优化

显存优化是破解大模型推理性能瓶颈的关键突破口，其核心在于高效管理随序列长度线性增长的 KV Cache，避免显存浪费、碎片化与容量溢出。传统推理引擎为每个请求按预设最大上下文长度静态分配连续显存空间存储 KV Cache，导致实际短请求场景下显存利用率低下，并在动态批处理中产生严重碎片问题。PagedAttention 技术借鉴操作系统虚拟内存分页机制，将 KV Cache 划分为固定大小的逻辑

页，并将其映射至非连续的物理显存页，实现按需分配与灵活复用，可显著提升显存利用率，实验表明 vLLM 采用该技术后在相同硬件下可支持的并发请求数提升 3 倍以上[10]。Prefix Caching（如 RadixAttention）通过识别并缓存系统提示词、历史对话等共享前缀的 KV Cache，在多轮对话或批量相似请求场景下避免重复计算，有效降低首 Token 延迟并提升整体吞吐[11]。此外，面对上下文窗口持续扩展（如百万 Token 级别）带来的显存压力，KV Cache 卸载（Offloading）策略将非活跃或低频访问的缓存数据迁移至 CPU 内存、SSD 甚至远程存储，构建“显存-内存-存储”的多级缓存体系，在保障推理连续性的同时显著扩展可处理上下文长度，并降低对高成本 GPU 资源的依赖[12]。



来源：https://zhuanlan.zhihu.com/p/693556044

图 7 KV Cache 前缀缓存与复用

2.计算优化

计算优化聚焦提升硬件计算单元的利用率，通过减少冗余浮点运

[10] Wu, W. L., et al. "vLLM: Easy, Fast and Cheap LLM Serving at Scale." arXiv:2305.14283 (2023). https://arxiv.org/abs/2305.14283

[11] SGLang Team. "RadixAttention: Efficient Attention with Prefix Caching." SGLang Documentation, 2024. https://sgl-project.github.io/

[12] "Mooncake: A Unified KV Cache Management System for LLM Serving." arXiv:2406.08983 (2024). https://arxiv.org/abs/2406.08983

算（FLOPs）与内存访问开销，突破大模型推理中的计算与带宽瓶颈。

一是，通过算子融合降低访存消耗。算子融合技术将多个细粒度算子（如矩阵乘法、加法、激活函数）合并为复合算子，减少内核启动次数和中间结果的显存读写操作。通过将中间张量保留在高速片上缓存或寄存器中，算子融合可显著降低内存带宽消耗并提高运算吞吐[13]。FlashAttention 通过重新组织注意力计算流程，将完整的注意力操作过程融合为单一 CUDA 内核，在执行中采用 I/O 感知和分块策略，避免中间结果矩阵在显存中的频繁读写，从而显著提升注意力计算性能[14]。其升级版本 FlashAttention-2 进一步通过优化线程块划分与减少非 GEMM 操作，在 NVIDIA A100 GPU 上实现 50–73%的 FLOPs 利用率[15]。**二是，**结合硬件进行内核级优化（Kernel Optimization）。通过结合特定硬件架构（如 NVIDIA Tensor Core）定制高频算子、性能瓶颈算子，实现最大化计算吞吐。DeepGEMM 代表了通用矩阵乘法（GEMM）在硬件协同优化方向的先进实践。DeepGEMM 针对 NVIDIA Hopper 架构（H100）进行深度优化，通过 FP8 低精度计算充分利用 Tensor Core 指令集，在保持模型精度可控的前提下，显著提高矩阵运算吞吐率[16][17]。

3.并行加速

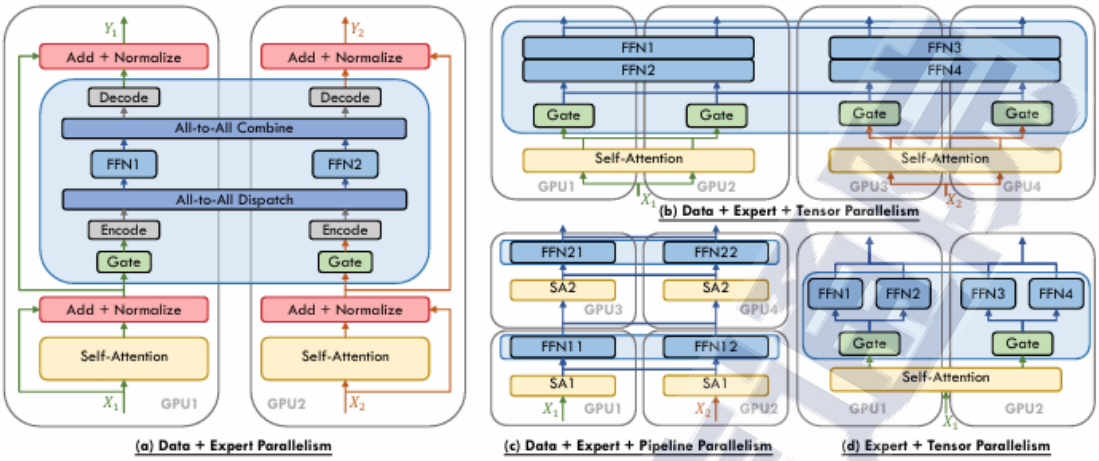
并行加速策略通过多维度并行方法提升系统吞吐率、降低显存压

[13] NVIDIA. TensorRT-LLM Developer Guide: Operator Fusion. https://docs.nvidia.com/deeplearning/tensorrt/llm/
[14] Dao, T., et al. "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness." Advances in Neural Information Processing Systems (NeurIPS), 2022. https://arxiv.org/abs/2205.14135
[15] Dao, T., Fu, D. Y., Zhao, Y., et al. (2023). FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691. https://arxiv.org/abs/2307.08691
[16] DeepGEMM Official Documentation. (2024). FP8 GEMM Optimization for Hopper Architecture. https://www.deepep.org/en/deepgemm
[17] DeepSeek AI. (2024). DeepGEMM GitHub Repository. https://github.com/deepseek-ai/DeepGEMM

力，优化计算资源利用率。不同并行策略各有侧重，需根据模型规模、模型架构、上下文长度及硬件条件组合使用，实现整体性能的最优化。**数据并行**（Data Parallelism, DP）将模型副本部署到多个设备上，通过划分输入数据实现并行处理，从而提升系统整体吞吐量。该策略适用于处理大规模并发请求，但对单卡内存压力无直接缓解作用。**张量并行**（Tensor Parallelism, TP）通过对模型内部参数矩阵按行或列进行切分，将单个算子的计算分布到多个设备上，解决单层模型参数过大无法被单个 GPU 承载的问题。然而，TP 通常伴随较高的通信开销，需要通过高效的通信策略和分块矩阵乘法进行优化。**流水线并行**（Pipeline Parallelism, PP）按模型层级划分阶段，构建计算流水线。由于阶段之间存在依赖关系，后一阶段必须等待前一阶段完成首个微批次（micro-batch）数据的计算才能开始，从而产生流水线“气泡”，影响部分设备的利用率。此外，随着模型架构、推理场景的发展，新的并行策略应运而生。**专家并行**（Expert Parallelism, EP）主要用于 MoE 模型，通过将不同专家分配到不同 GPU 上，并优化负载均衡与通信效率，实现对大规模专家网络的高效计算，同时降低单卡显存占用。**序列并行**（Sequence Parallelism, SP）沿序列长度维度划分输入激活值，在张量并行基础上进一步降低显存压力，对长上下文推理尤为关键。序列并行可有效减少激活值的内存占用，并提升长序列处理效率。

在实际部署中，混合并行策略成为主流方案，通过组合 DP、TP、PP、EP 和 SP，可充分发挥各策略优势，实现吞吐率、显存占用及通信开销的综合优化。这类多维度并行方法是支撑超大规模模型高效推

理的关键技术手段[18][19]。



来源：arXiv:2407.06204v3

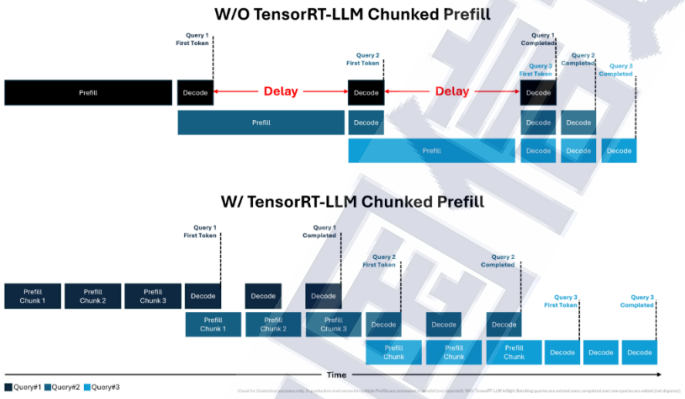
图 8 MoE 模型的多种并行策略示意图

4.批调度优化

批调度优化旨在通过智能组织和处理推理请求队列，尤其应对推理解码阶段输出长度不定的特性。传统静态批处理（Static Batching）需等待批次中最慢请求完成，在大模型生成长度不定情况下，将引发资源浪费。**动态批处理**（Dynamic Batching）通过将多个推理请求组合成统一批次计算，根据实时请求队列动态调整批次大小，在等待时间与吞吐量之间实现更好的平衡。然而，当请求的序列长度差异较大时，可能导致内存利用率不均衡，进而影响系统性能。为了进一步处理批次内序列长度差异较大的情况，**连续批处理**（Continuous Batching）采用迭代级调度，在每轮迭代中动态决定批次大小，请求一旦完成生成便立即从批次中移除，并动态地将等待队列中的新请求加入批次，从而显著提升了 GPU 的利用率和整体吞吐量。此外，针对长上下文

[18] A Survey on Mixture of Experts in Large Language Models https://arxiv.org/pdf/2407.06204
[19] DeepSpeed. https://www.deepspeed.ai/tutorials/mixture-of-experts/

推理中存在的显存瓶颈与延迟挑战，**分块预填充**（Chunked-Prefills）将长输入序列分割为多个较小块按顺序处理，而非一次性处理整个序列，在适当边界处允许块处理与后续计算重叠，同时缓存已处理块的键值状态，避免重复计算。这种方法平衡了显存使用与计算效率，使长序列推理更加可行，同时降低峰值显存使用和首 Token 等待时间[20]，是 PD（Prefill-Decode）分离的引擎级实现。



来源：Streamlining AI Inference Performance and Deployment with NVIDIA TensorRT-LLM Chunked Prefill

图 9 Chunked-Prefill 过程示意图

（三）系统层面

系统层是大模型推理体系全局控制与服务执行的核心，负责在跨节点、跨实例、跨资源的复杂环境中实现推理任务的高效协同与稳定交付。其核心目标在于在规模化场景下实现高性能、低时延与成本可控的服务化推理。**不同于引擎层侧重单实例执行效率，系统层优化的边界扩展到分布式推理架构与服务化体系，关注整体吞吐、时延稳定性与成本平衡。**其关键技术包括结合模型、服务特性的分布式系统架

[20] NVIDIA Developer Blog. (2024). Streamlining AI Inference Performance and Deployment with NVIDIA TensorRT-LLM Chunked Prefill. https://developer.nvidia.com/blog/streamlining-ai-inference-performance-and-deployment-with-nvidia-tensorrt-llm-chunked-prefill/

构优化、分布式调度策略优化以及高性能存储等。

1.PD 分离架构

预填充-解码（Prefill-Decode, PD）分离式推理架构已成为业界主流优化方案。大模型推理一般由预填充（Prefill）和解码（Decode）两阶段构成，其中预填充阶段是计算密集型（compute-bound）对算力需求高，容易迅速使 GPU 达到饱和；解码阶段是存储密集型（memory-bound）对显存需求高，在大批量（batch size）请求下才可充分利用计算资源，同时受到带宽限制。传统方式通常直接将推理服务部署到集群中，使得 PD 两阶段在同一节点上执行，引发两阶段资源争夺、并行策略互相掣肘难以优化，进一步导致资源利用率低、服务性能差、系统构建成本高等问题[21][22]。

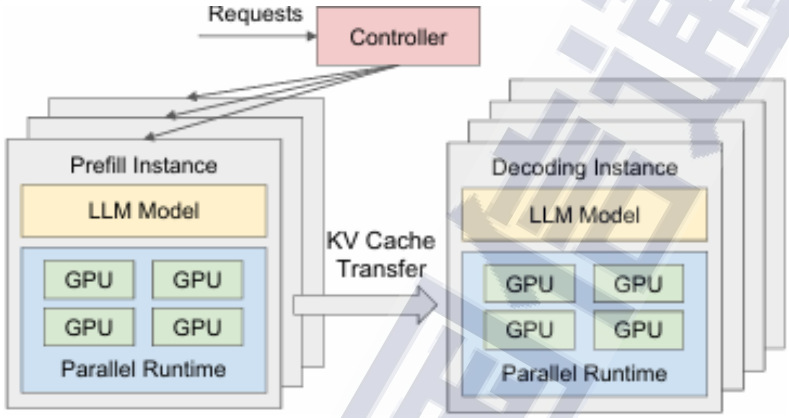
PD 分离架构通过结构性解耦，将推理过程拆分为预填充与解码两个独立阶段，从根本上缓解计算资源与内存带宽的冲突。其核心思路是“阶段解耦、专用部署、并行调度”：预填充阶段可部署在高算力 GPU 节点以集中执行大规模矩阵运算；解码阶段则运行在优化了内存访问路径的节点上，以低延迟方式持续生成 Token。该设计不仅提升了各阶段资源利用率，还显著降低了请求延迟，并提升了系统的并发处理能力[23]。**在该架构中，KV Cache 是连接两阶段的关键枢纽。**预填充阶段生成的 KV Cache 可被多个解码实例复用，尤其在不同请求

[21] Throughput is Not All You Need: Maximizing Goodput in LLM Serving using Prefill-Decode Disaggregation https://hao-ai-lab.github.io/blogs/distserve/

[22] 中金 | AI 十年展望（二十）：细数 2024 大模型底层变化，推理优化、工程为王 https://mp.weixin.qq.com/s/tY3pxGpg-WK70ySOgkkiRQ

[23] Zhong, Y., et al. (2024). DistServe: Disaggregating Prefill and Decoding for Goodput-Optimized Large Language Model Serving. OSDI 2024. https://www.usenix.org/system/files/osdi24-zhong-yinmin.pdf

共享相似 Prompt 前缀时，可避免重复计算历史上下文[24]。此外，通过缓存持久化技术，系统能够在多轮对话中跨轮次复用缓存，有效降低重复推理开销。动态缓存管理策略进一步根据访问频率、优先级与显存状态自动决定缓存的分配与淘汰策略，确保有限显存资源下的最高命中率。



来源：arXiv:2401.09670v3

图 10 PD 分离架构示意图

尽管 PD 分离架构在性能与扩展性方面表现优秀，但仍面临三大核心挑战。一是传输开销，大量 KV Cache 跨节点传输可能引入显著延迟。**二是显存压力，**解码阶段需同时维护大量并发请求的 KV Cache，而 HBM/DRAM 等高带宽存储资源有限。**三是缓存协同复杂，**预填充生成与解码访问需精准匹配，否则缓存失配或冗余访问将削弱优化效果[25]。为应对上述问题，业界提出多层次优化路径：在通信层采用 RDMA/NVLink 等高速互联技术，并结合 KV Cache 量化与按层流式传输策略以降低传输延迟；在存储层通过 HBM-DRAM-SSD 分级架

[24] Chen, W., He, S., Qu, H., Zhang, R., Yang, S., Zheng, Y., Huai, B., Chen, G. (2025). IMPRESS: An Importance-Informed Multi-Tier Prefix KV Storage System for Large Language Model Inference. FAST 2025. https://www.usenix.org/system/files/fast25-chen-weijian-impress.pdf

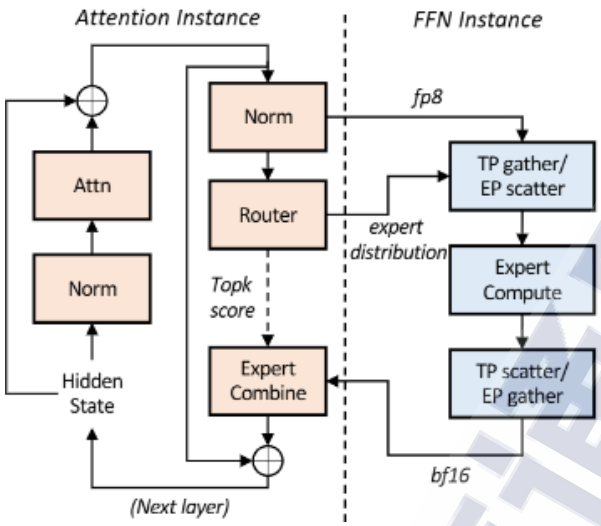
[25] Li, W., Jiang, G., Ding, X., Tao, Z., Hao, C., Xu, C., Zhang, Y., Wang, H. (2025). FlowKV: A Disaggregated Inference Framework with Low-Latency KV Cache Transfer and Load-Aware Scheduling. arXiv. https://arxiv.org/abs/2504.03775

构与缓存复用技术减轻显存占用；在调度层实现预填充与解码异步重叠执行、计算与传输并行，从而全面提升系统推理效率及扩展能力。

2.AF 分离架构

在 MoE 架构模型的推理场景中，注意力（Attention）层与前向反馈（Feedforward）层呈现出显著的计算特征差异：前者为访存密集型，依赖 KV Cache 的频繁访问；后者为计算密集型，主要执行大规模矩阵乘运算。这种计算异构性使得在同一 GPU 上混合执行两类操作时，资源利用率往往受限，GPU 计算单元与显存带宽难以同时饱和[26]。为应对此问题，业界提出了 AF 分离（Attention–Feedforward Disaggregation, AFD）架构，继承了 PD 分离“按任务性质分治优化”的系统思想。AF 分离通过在系统层面将 Attention 模块与 Feedforward 模块拆分至不同计算节点，使两类任务可独立优化并并行执行。Attention 节点（A 节点）部署在具备高显存带宽与大容量缓存的 GPU 上，用于处理 KV Cache 与注意力计算；Feedforward 节点（F 节点）则部署在高算力、性价比更优的 GPU 上以执行 Dense 计算。该设计消除了计算与访存冲突，实现了异构资源的协同最优利用。

[26] Step-3 is Large yet Affordable: Model-system Co-design for Cost-effective Decoding https://github.com/stepfun-ai/Step3/blob/main/Step3-Sys-Tech-Report.pdf



来源：Step-3 is Large yet Affordable

图 11 Step-3 的 AF 分离架构

AF 分离架构支持异构部署与独立扩展。当模型输入长度变化或系统 SLO 收紧时，A 与 F 节点可分别按需扩展规模以维持目标延迟与吞吐的平衡。可分级伸缩的特性使得 AF 分离在多租户、长上下文、高并发场景下具有显著优势。**此外，从模型系统协同设计的角度看，AF 分离亦为模型结构优化提供了理论依据。**例如，分析表明 FFN 层稀疏度与通信带宽呈正相关关系，合理引入低秩映射可在不增加通信负担的情况下提高稀疏度[27]，从而实现更优的性能–精度–成本平衡。

AF 分离架构标志着大模型系统架构从 PD 分离为代表的阶段分治，迈向 AF 分离为代表的模块分治。通过计算性质解耦实现资源利用率最大化与系统级最优推理吞吐，为未来 AI 推理系统的异构计算调度与模型共设计奠定基础。

3.系统调度策略

[27] Step-3 is Large yet Affordable: Model-system Co-design for Cost-effective Decoding https://github.com/stepfun-ai/Step3/blob/main/Step3-Sys-Tech-Report.pdf

调度策略作为系统架构的智能中枢，协调着整个推理系统的运行效率。通过智能化请求分发与资源分配，使推理系统在满足 SLO 要求的同时，最大化算、网、存的资源利用率，最小化成本与长尾延迟，并能动态应对请求异构性与节点故障[28]。**现代调度系统主要围绕请求调度和资源调度两个维度展开优化。**请求调度负责将用户请求智能分配给最合适的计算实例，综合考虑请求特性、实例负载和系统状态等多重因素。资源调度则关注硬件资源的动态分配与再平衡，根据工作负载变化实时调整计算、显存和带宽资源的分配。在 MoE 模型场景中，资源调度尤为重要，需要根据专家激活模式动态调整各专家的计算资源，以充分发挥 MoE 架构的效能。

缓存亲和性调度策略通过分析请求特征，将相似请求路由到具有对应 KV Cache 的实例处理，极大提升了请求处理效率。这种策略与 PD 分离架构完美契合，使得预填充阶段生成的 KV Cache 能够在多个解码请求间高效共享。**负载感知调度策略**通过实时监控各推理实例的 GPU 利用率、显存使用率、请求队列长度等关键指标，实现系统负载的动态均衡，避免部分实例过载而其他实例闲置的情况。**故障感知与容错调度策略**则通过多层次健康检查和自动故障转移机制保障服务连续性，结合请求重试和结果缓存策略，确保在部分组件故障时系统仍能维持基本服务能力。这些调度策略共同作用，有效解决了实例碎片、缓存优化、路由效率等核心问题，为大模型推理服务提供了可靠的系统级保障。

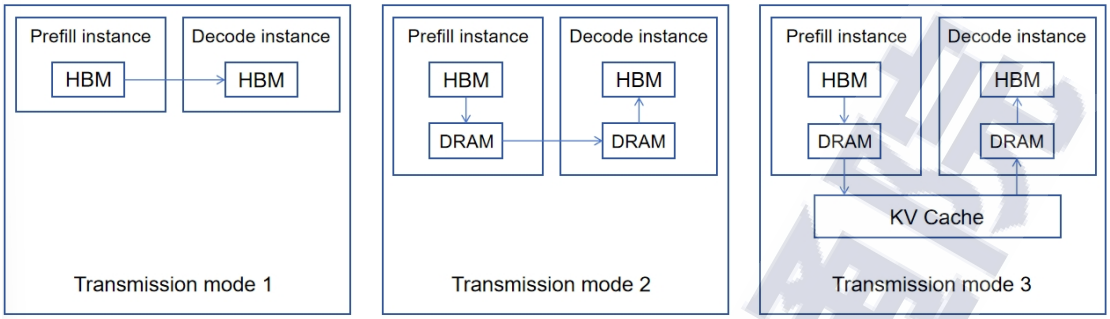
[28] Llumnix: Dynamic Scheduling for Large Language Model Serving https://doi.org/10.48550/arXiv.2406.03243

4.高性能存储

随着模型参数与上下文长度的快速增长，KV Cache 作为模型的“工作记忆”，其存储开销呈平方级增长，对 GPU 显存构成巨大压力。传统单一依赖高带宽内存（HBM）的方案，在面临长上下文、高并发推理任务时，常因显存容量限制而导致吞吐下降、成本激增。为此，构建智能、高效的多级存储体系，已成为保障推理服务性能与经济效益的必经之路。

为应对显存资源稀缺性与成本挑战，动态多级长记忆存储架构被普遍使用。业界通常使用“HBM-DRAM-SSD”构成的三级动态存储架构。该体系将 KV Cache 依据访问特性与生命周期，智能调度于不同层级的存储介质中，其中高频访问的"极热"数据驻留于高带宽的 HBM，中低频的"热"或"温冷"数据则迁移至容量更大、成本更低的 DRAM 和外置存储，实现性能与容量的最优平衡。这一架构设计旨在突破物理显存容量限制，支持更长上下文与更大批次并发，同时，通过低成本、高性能、大容量的外置存储有效扩展系统数据容量，优化“内存墙”和“容量墙”，进而显著降低系统整体成本。**多级存储架构的有效性依赖于一系列关键技术的深度协同。**系统通过实时监控访问频率、时序局部性等指标，精准识别 KV Cache 的"冷热"状态，并借助后台异步任务，在各级存储间执行数据的智能换入换出。为最大化存储效率，系统引入前缀缓存等复用机制，当不同请求共享相同提示前缀时，直接复用已生成的 KV Cache，避免重复计算。同时，通过 RDMA 与加速卡直通存储等底层技术，构建 GPU 与存储设备间的直接数据

通道，消除 CPU 拷贝瓶颈，确保数据迁移的高效性。



来源：华为技术有限公司

图 12 PD 分离中的三种典型存储架构

PD 分离架构为存储系统创造了独特的优化空间。在 PD 分离架构场景下，多级存储体系为 Prefill 阶段生成的 KV Cache 提供了跨节点的缓存共享与持久化能力，使得解码节点能够按需、低延迟地获取缓存。同时，通过将非活跃的历史缓存数据卸载至 DRAM 和外置存储，显著缓解了解码节点对昂贵 HBM 的依赖，使其支持更高并发度。存储系统还可与调度器协同，实现 KV Cache 的异步流式传输与智能预取，有效隐藏数据访问延迟。

尽管多级存储架构优势显著，其落地仍面临诸多挑战并驱动着技术持续演进。在缓存数据于多层级、多节点间动态迁移时，如何确保解码阶段访问到的 KV Cache 始终保持完整性与一致性，是维护推理正确性的关键挑战。同时，HBM、DRAM、SSD 等存储器在带宽与延迟上存在数量级差异，系统需要精细设计数据放置与迁移策略，避免低速 I/O 成为整个推理管道的性能瓶颈。未来，多级 KV Cache 存储需要与并行文件系统、数据编织等跨域数据调度技术深度融合，实现跨集群的缓存资源池化与统一管理，为超大规模模型推理服务系统奠

定坚实基础。

四、大模型推理优化应用实践

（一）前期：聚焦平台功能完备

早期阶段，产业界普遍聚焦于构建功能完备的推理服务平台，以实现从模型调优到部署推理、服务化交付的全流程贯通。该阶段主要通过体系化工具集与云化环境，打通大模型在生产场景落地的基础环节，为后续性能优化与协同架构演进提供支撑。

一是，平台体系化能力初步形成，构建了模型推理的基础运行框架。早期平台重点在于整合模型管理、部署调度、接口服务、运行监控等核心功能，形成从“训练输出模型”到“推理提供服务”的闭环机制。**二是，平台功能从通用大语言模型部署扩展至多场景、多模态服务。**随着企业对生成式 AI 的应用需求多元化，平台建设逐步由单点部署能力向多类型推理任务支撑扩展。典型特征包括支持文本生成、图像生成、语音理解等多模态推理接口，以及在线推理、离线批处理、RAG（检索增强生成）等多样化运行模式。这类多模态与多场景的扩展，使平台从“功能完备”进一步迈向“业务可复用”，进一步推动生成式 AI 应用的普及。**三是，平台能力向标准化与易用化方向深化。**在功能逐步完备的基础上，平台建设进一步强调标准接口、低门槛使用与可扩展生态。云服务厂商普遍提供一键部署、弹性伸缩、监控可视化等特性，显著降低了企业构建推理服务的门槛。

（二）现状和趋势：方案迭代，从单点优化走向系统优化

1.单点优化：压缩工具与推理引擎是两大关键方向

在大模型推理优化的早期阶段，产业实践主要集中于模型压缩与推理引擎两大方向，形成了以单点性能优化为核心的技术体系，为后续系统协同优化奠定基础。这类方案具有轻量化、易实现、落地快的特点，能够在较低工程复杂度下带来性能提升，但优化空间有限、难以应对多场景复杂需求。

模型压缩工具核心目标是在可接受精度下降范围内，尽可能降低模型的存储与计算代价，以增强模型的可部署性以及成本效率。一是，压缩技术逐步演进为面向大模型特性的精细化轻量化方案。Meta 采用的 GPTQ[29]与 MIT 提出的 AWQ[30]量化方法在权重量化过程中引入误差补偿、激活感知等机制，实现量化后精度保证的同时，降低显存占用并提升推理速度。**二是，**压缩工具化与平台化集成，成为推理优化体系中标准化模块。微软推出的 DeepSpeed Compression[31]将量化、剪枝、蒸馏功能封装为统一 API 接口，支持与 DeepSpeed-Inference 等框架联动；英伟达在 TensorRT Toolkit 中集成高精度量化与校准模块，核心通过校准技术（如 KL 散度校准、最小化量化误差校准），在 INT8/INT4 低精度下将精度损失控制在 1%-3%以内，大幅降低显存占

[29] Frantar, Elias et al. “GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers.” arXiv:2210.17323 https://arxiv.org/abs/2210.17323

[30] Lin, Ji et al. “AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration.” arXiv:2306.00978 https://arxiv.org/abs/2306.00978

[31] Microsoft Research Blog. “DeepSpeed Compression: Efficient Compression for Large Models.” (2023) https://www.microsoft.com/en-us/research/blog/deepspeed

用与计算开销。阿里云 PAI 平台推出 PAI-Blade、PAI-EasyDistill、PAI-Model Gallery 等模型轻量化组件，可提供丰富的模型压缩功能支持；百度千帆大模型平台集成 PaddleSlim 压缩工具，实现结构化剪枝、蒸馏与动态量化的统一管理。压缩工具已从实验性算法转变为产业级标准组件，成为推理优化流程的重要入口。**三是，**与模型、硬件协同的高效压缩方案正成为新的趋势。英伟达在 Hopper 架构 GPU 中引入对 FP8 量化的原生支持，并通过 TensorRT-LLM 自动调用低精度算子实现算力加速[32]。

推理引擎作为大模型推理体系中连接模型与算力的执行核心，聚焦于单实例或轻量集群内的计算与内存优化，是推理优化早期阶段单点性能提升的主要抓手。一是，通用推理引擎经历“多点爆发-功能富集-逐步收敛”的演化轨迹。从前期学术机构、初创企业、大型科技公司、技术社区等推出的几十种推理引擎，逐渐收敛至 vLLM（技术社区）、SGLang（技术社区）、llama.cpp（技术社区）、Text Generation Inference（HuggingFace）、DeepSpeed-FastGen（Microsoft）、TensorRT-LLM（NVIDIA）等几个主流推理引擎，其中 vLLM、SGLang 因兼具推理优化特性丰富、功能更新迅速、社区支持充分、分布式支持能力强、二次开发支持性好、多硬件支持好、与 DeepSeek/MoE 等主流模型深度协同等优势，其产业应用普及性尤为突出[33]。**二是，**围绕服务化场景特点，主流推理引擎不断纳入新特性，同时也衍生出一系列差

[32] https://www.nvidia.cn/data-center/technologies/hopper-architecture/
[33] A Survey on Inference Engines for Large Language Models: Perspectives on Optimization and Efficiency https://www.arxiv.org/pdf/2505.01658

异化、专用化推理引擎。LMDeploy 在语言模型之外还关注视觉语言模型（VLs），主要围绕 GPU 进行深度优化，适用于企业级大规模、高稳定性、高性能应用和实时系统。SGLang 对于分布式部署、长文本场景支持优异，尤其自 DeepSeek-V2 开始 SGLang 与 DeepSeek 深度整合，在性能测试中对于高并发、低时延等差异化场景均能完美胜任，产业界对 SGLang 的关注迅速提升[34]。**三是，**随着 MoE 模型架构逐渐成为主流趋势，主流大模型框架如 vLLM、DeepSpeed[35]等均强化了对 MoE 特性的支持，同时产业界也推出了一批聚焦 MoE 训推的 AI 框架，如清华的 KTransformers 通过优化底层算子、引入专家延迟机制(Expert Deferral)等创新技术，实现了 CPU 与 GPU 的高效协同。此外，DeepSeek 也推出了为 MoE 架构中专家并行（EP）定向优化的 DeepEP 通信库[36]。

[34] 大型语言模型（LLM）推理框架的全面分析与选型指南（2025 年版）
https://blog.csdn.net/Wufjsjjx/article/details/146043448
[35] Getting Started with DeepSpeed-MoE for Inferencing Large-Scale MoE Models
https://www.deepspeed.ai/tutorials/mixture-of-experts-inference/
[36] DeepEP https://github.com/deepseek-ai/DeepEP

Frameworks	Organization	Release Date	Open-Source Support†	GitHub # Stars (Rate)	GitHub Star	GitHub Commit	Supported Models‡	Docs*	User Forum** S	User Forum** F	User Forum** M
Ollama [194]	Community (Ollama)	Jun. 2023	✔	136K (209.6)				✔	✔	✘	✔
llama.cpp [82]	Community (gml.ai)	Mar. 2023	✔	77.6K (102.6)				✔	✘	✘	✘
vLLM [125]	Academic (vLLM Team)	Feb. 2023	✔	43.4K (55.2)				✔	✔	✔	✔
DeepSpeed-FastGen [102]	Big Tech (Microsoft)	Nov. 2023	✔	37.7K (72.6)				✔	✘	✘	✔
Unsloth [244]	Startup (unsloth AI)	Nov. 2023	▲	36.5K (74.1)				✔	✔	✔	✘
MAX [179]	Startup (Modular Inc.)	Apr. 2023	▲	23.8K (33.7)				✔	✔	✔	✔
MLC LLM [177]	Community (MLC-AI)	Apr. 2023	✔	20.3K (28.8)				✔	✔	✘	✘
llama2.c [21]	Community (Andrej Karpathy)	Jul. 2023	✔	18.3K (29.1)				✘	✔	✘	✘
bitnet.cpp [251]	Big Tech (Microsoft)	Oct. 2024	✔	13.6K (53.0)				✘	✘	✘	✘
SGLang [295]	Academic (SGLang Team)	Jan. 2024	✔	12.8K (28.4)				✔	✔	✘	✔
LitGPT [145]	Startup (Lightning AI)	Jun. 2024	✔	12.0K (16.6)				✔	✔	✘	✔
OpenLLM [30]	Startup (BentoML)	Apr. 2023	▲	11.1K (15.5)				✘	✔	✘	✘
TensorRT-LLM [191]	Big Tech (NVIDIA)	Aug. 2023	▲	10.1K (16.1)				✔	✘	✔	✔
TGI [110]	Startup (Hugging Face)	Oct. 2022	✔	10.0K (11.0)				✔	✘	✔	✘
PowerInfer [227]	Academic (SJTU-IPADS)	Dec. 2023	✔	8.2K (17.2)				✘	✘	✘	✘
LMDeploy [162]	Startup (MMRazor/MMDeploy)	Jun. 2023	✔	6.0K (9.1)				✔	✔	✘	✘
LightLLM [142]	Academic (Lightllm Team)	Jul. 2023	✔	3.1K (5.0)				✔	✔	✘	✘
NanoFlow [300]	Academic (UW Efeslab)	Aug. 2024	✔	0.7K (3.5)				✘	✘	✘	✘
DistServe [297]	Academic (PKU)	Jan. 2024	✔	0.5K (1.2)				✘	✘	✘	✘
vAttention [206]	Big Tech (Microsoft)	May. 2024	✔	0.3K (1.0)				✘	✘	✘	✘
Sarathi-Serve [9]	Big Tech (Microsoft)	Nov. 2023	✔	0.3K (0.6)				✘	✘	✘	✘
Friendli Inference [71]	Startup (FriendliAI Inc.)	Nov. 2023	✘	–	–	–		✔	✘	✘	✔
Fireworks AI [67]	Startup (Fireworks AI, Inc.)	Jul. 2023	✘	–	–	–		✔	✔	✘	✘
GroqCloud [89]	Startup (Groq Inc.)	Feb. 2024	✘	–	–	–		✘	✔	✘	✔
Together Inference [239]	Startup (together.ai)	Nov. 2023	✘	–	–	–		✔	✔	✘	✘

† ▲indicates partial open-source support, ‡ Each square represents 50 models (Mar. 2025)
* Indicates the level of detail of the document (✔: Simple, ✔: Moderate, ✔: Detail),
** S refers for social networking services (Discord/Slack), F refers for discussion forums (private forums/reddit), and M refers for meetups

来源：A Survey on Inference Engines for Large Language Models: Perspectives on Optimization and Efficiency

图 13 25 种大语言推理引擎概况对比

2.协同优化基础：PD 分离开启“模型-架构-场景”协同优化新篇章

随着 PD 分离式推理架构逐渐成熟，场景落地显著加速。2024 年陆续推出了 DistServe（北大&USCD）、Splitwise（微软）、TetriInfer（华为云）和 MemServe（华为云）等 PD 分离式推理架构方案[37]。

2024 年，DistServe 提出了基础的 PD 分离架构，详细分析了 PD 分离架构对于大模型推理的适配性。该方案虽未涉及异构资源、复杂调度等针对场景特性的策略设计，但其开源代码仍为后续 PD 分离方向的方案演进奠定了基础[38]。2024 年 5 月，Splitwise 初步验证了 PD 分离方案在异构硬件资源运行，可在保证在性能要求下实现成本压缩；

[37] 大模型推理分离架构五虎上将 https://mp.weixin.qq.com/s/g7lq4IcJ4-etkh9XV8Giig
[38] Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving https://arxiv.org/abs/2401.09670

同时，提出了针对节点内、节点间的调度策略以及 KV Cache 传输策略的优化方向[39]。同年，TetriInfer 从工业落地角度进一步细化系统模块划分与调度策略，从实际应用中常见的分布式架构角度，提出结合 P/D 实例负载的调度策略，以保障负载均衡；同时，提出 P/D 实例转化机制，提升内部可扩展性[40]。MemServe 则从实际应用中常见的请求调度策略角度，提出基于全局请求的 Global Prompt Tree 管理机制，通过构建弹性内存池配合全局调度器，管理集群中的 KV Cache 与路由方式，突破当时传统架构主要聚焦实例内的局限性[41]。

表 1 大模型推理 Prefill-Decode 阶段对比[42]

特性	**Prefill 阶段**	**Decode 阶段**
操作	一般对应提示词处理	一般对应解码生成
处理单元	整个提示词 (N 个词元)	单个新词元 (1 个词元)
计算模式	并行计算	串行（自回归）计算
计算复杂度	O(n^2)，n 为 prompt 长度	O(1)/token
瓶颈	计算密集型(Compute-Bound)	内存带宽密集型(Memory-Bound)
主要操作	矩阵-矩阵运算 (GEMM)	矩阵-向量运算 (GEMV)
核心指标	首字输出延迟 (Time to First Token, TTFT)	每字生成时间 (Time Between Tokens, TBT)

3.协同优化趋势一：以 KV Cache 为核心的架构优化

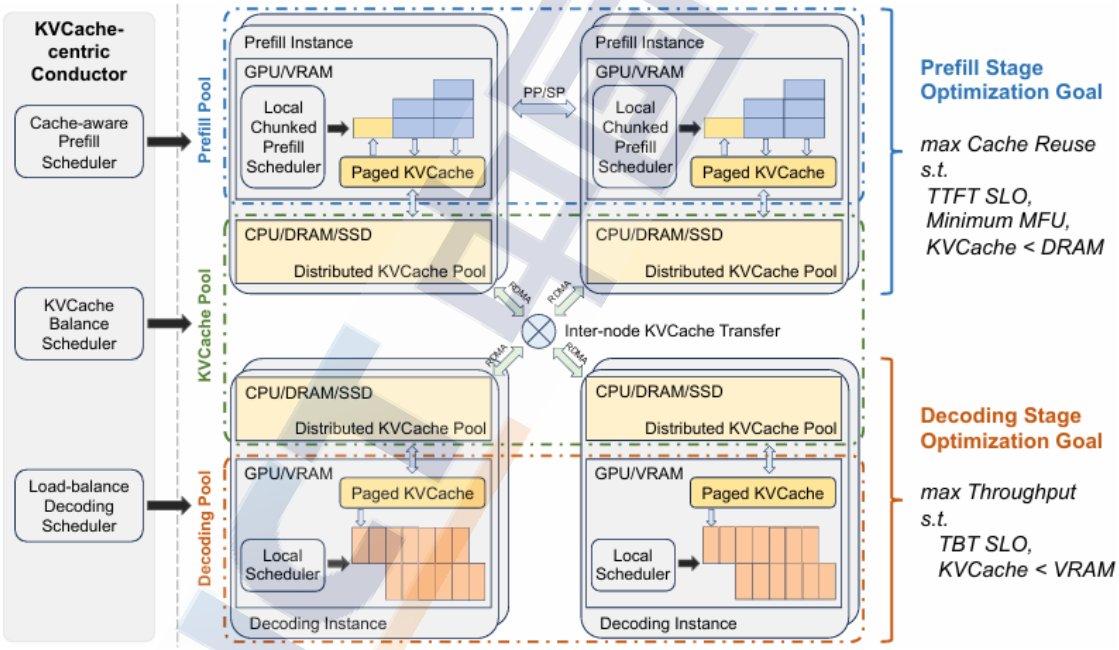
当前，以 Mooncake（月之暗面）、Dynamo（英伟达）、UCM（华为）等为代表的工业级方案迭出，以上方案均在 PD 分离架构基础上，

[39] P. Patel, E. Choukse, C. Zhang, A. Shah, Í. Goiri, S. Maleki, and R. Bianchini, "Splitwise: Efficient Generative LLM Inference Using Phase Splitting," arXiv:2311.18677v2, May 2024. https://arxiv.org/pdf/2311.18677v2
[40] Inference without Interference: Disaggregate LLM Inference for Mixed Downstream Workloads https://arxiv.org/abs/2401.11181
[41] MemServe: Context Caching for Disaggregated LLM Serving with Elastic Memory Pool https://arxiv.org/abs/2406.17565
[42] 大模型 PD 分离部署万字长文深度解析 https://mp.weixin.qq.com/s/uj4CQEFUW-LIDq3p_ebFuw

提出针对 KV Cache 的“存储-计算-调度”方案，持续推动以 KV Cache 为核心的推理系统路线演进。此外，也为请求调度、SLO 自适应方案、长上下文/流量过载等场景策略设计提供了多种思路。

1) Mooncake 架构方案

2025 年 2 月，月之暗面与清华联合阿里云、华为存储、面壁智能、趋境科技等共同发布了 Mooncake 开源项目，引起产业热烈轰动。该方案核心创新在于重构了推理服务的资源组织模式，通过全局调度器（Conductor）、P/D 集群及分布式 KV Cache 池（Mooncake Store）的协同架构，实现计算与存储的深度解耦。



来源：arXiv:2407.00079v4

图 14 Mooncake 架构图

一方面，Mooncake Store 作为统一存储池，创新性整合 AI 芯片原生的 HBM 以及集群闲置且成本更低的 CPU、DRAM、SSD 等存储资源，建立分级存储机制：高频访问的 KV Cache 块优先驻留 HBM

保障低延迟，历史缓存与低频数据动态下沉至 DRAM 与 SSD，通过“**以存换算**”策略大幅降低 HBM 资源占用，同时借助 RDMA 的高带宽、低延迟特性实现跨介质数据高效流转。**另一方面，**针对长上下文处理与过载场景等产业痛点，Mooncake 设计了多层次优化机制：在 Prefill 阶段采用分块流水线并行（CPP）技术，将超长输入令牌拆分后由多节点协同处理，结合计算与缓存操作的异步执行优化，有效降低长文本场景的 TTFT；全局调度器采用 KV Cache 感知策略，优先路由请求至存在最长可重用前缀缓存的节点，并通过启发式热点迁移机制复制高频缓存块，最大化跨会话缓存重用价值。针对流量波动场景，引入预测式早拒绝策略，基于历史负载数据预测 Decoding 节点未来容量，提前筛选无法满足 SLO 的请求，避免无效 Prefill 计算导致的资源浪费。

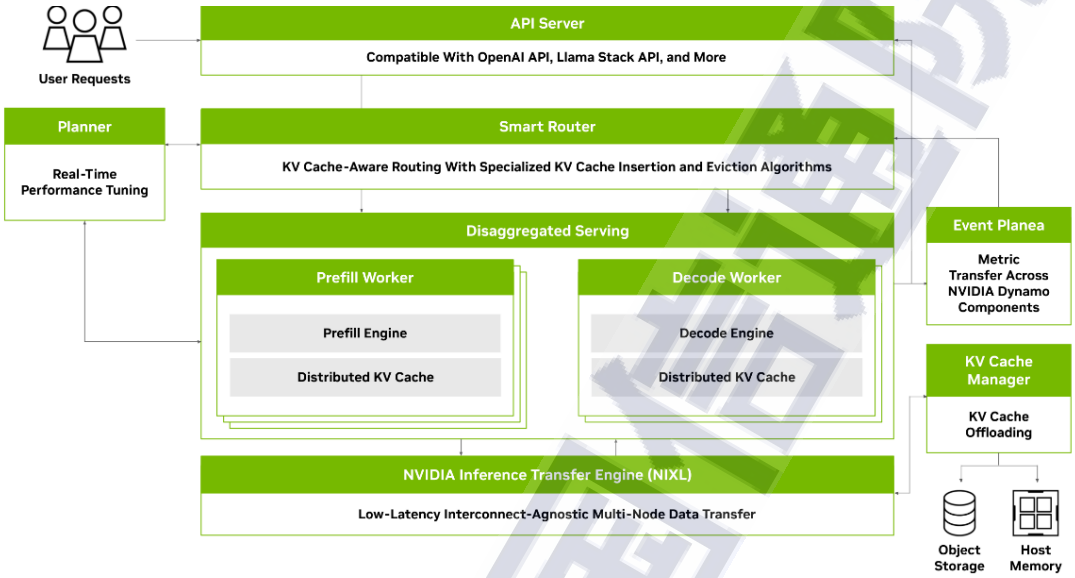
该方案已在 Kimi 大模型实现生产级规模化落地，承接其 80%以上的线上流量，可实现在真实工作负载下，系统有效吞吐量较基线方案提升 75%，100%满足 TBT 约束（vLLM 仅 57%），TTFT 分布与 vLLM 基本一致；在 16k~128k 输入长度的长上下文模拟场景中，吞吐量最高提升 525%，且严格满足 TTFT 和 TBT 的 SLO 约束[43]。Mooncake 提出的“以存换算”理念为产业后续方案迭代奠定坚实基础。

2）Dynamo 架构方案

2025 年 3 月，英伟达在 GTC 2025 推出 NVIDIA Dynamo 分布式推理加速项目。该方案的核心创新在于构建了模块化协同架构，通过

[43] Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving https://arxiv.org/pdf/2407.00079

SLO-based Planner、KV Cache 感知路由、多级存储管理及高效传输组件（NIXL）的深度联动，实现 Prefill 与 Decoding 阶段的柔性解耦，为分布式推理提供性能、成本与扩展性的最优平衡。



来源：https://developer.nvidia.cn/dynamo

图 15 Dynamo 架构图

一方面，针对分布式推理中的缓存复用、跨介质传输与成本优化等核心痛点，Dynamo 设计了多层次技术方案：KV Cache 感知路由器（Smart Router）通过实时追踪全集群 KV Cache 命中情况与节点负载，将请求路由至最优实例，显著降低 TTFT；分布式 KV Cache 管理器构建四级存储体系，将高频缓存驻留 GPU 显存，中低频数据下沉至 CPU 内存、本地 SSD 及外置存储，避免 KV Cache 重复计算，大幅降低高端显存占用成本；推理传输库（NIXL）兼容多种后端，提供高带宽、低延迟的跨介质数据传输支撑，为解耦架构提供高效通信保障。**另一方面，**系统设计“实时监控-智能决策-资源调度”机制，基于持续采集的 GPU 资源指标、请求特征（输入/输出序列长度，

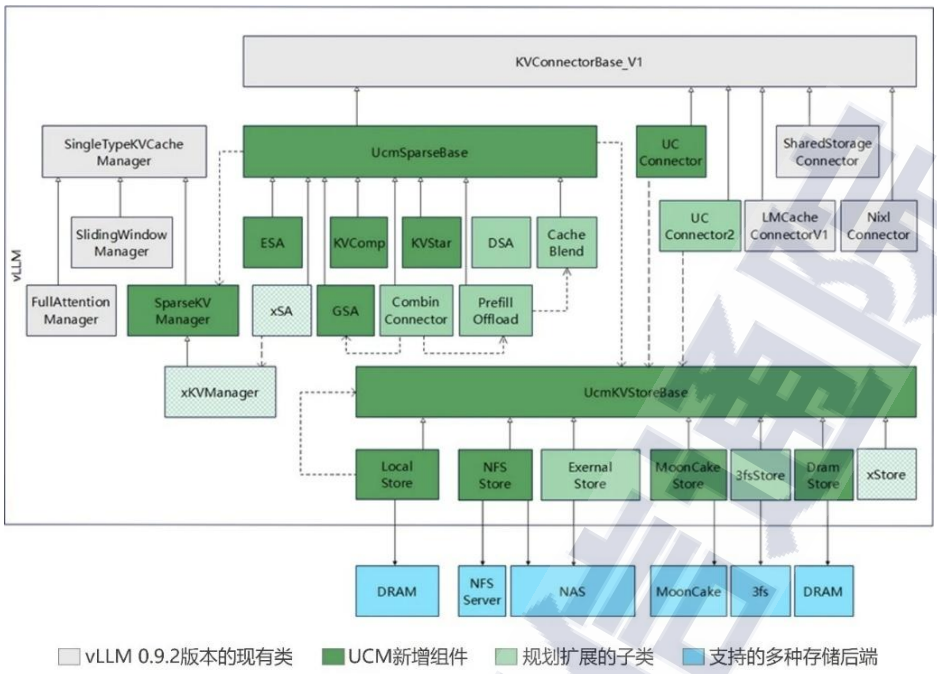
ISL/OSL）与 SLO 目标（TTFT/ITL 阈值）等信息，自动调整最优 PD 分离配置与并行策略，实现资源按需分配与自动扩缩容，大幅降低部署与运维复杂度。**此外，**Dynamo 构建了全栈兼容的技术生态，原生支持 TensorRT-LLM、vLLM、SGLang 三大主流推理引擎，支持多引擎并行评估与业务灵活适配[44][45]。

3) UCM 架构方案

2025 年 11 月，华为正式开源 UCM（Unified Cache Manager）推理记忆数据管理套件，以加速 AI 推理性能。UCM 以 KV Cache 多级缓存和推理记忆管理为中心，通过推理框架、算力、存储的三层协同，提供自适应全局 Prefix Cache、全流程稀疏注意力、后缀检索预测等关键推理加速技术，破解长序列推理效率低、成本高的难题，实现推理 TTFT 最高可降低 90%、系统吞吐最大可提升 22 倍和上下文窗口序列长度扩展 10 倍级。

[44] NVIDIA Dynamo, A Low-Latency Distributed Inference Framework for Scaling Reasoning AI Models https://developer.nvidia.com/blog/introducing-nvidia-dynamo-a-low-latency-distributed-inference-framework-for-scaling-reasoning-ai-models/

[45] NVIDIA Dynamo Platform https://developer.nvidia.cn/dynamo



来源：华为技术有限公司

图 16 UCM 架构图

一方面，UCM 在底层框架和机制上提供了多级缓存空间的分层管理与智能流动能力：实时数据存放在高性能缓存 HBM 中，短期数据存放在 DRAM 中，其他数据存放在共享的外置存储中，以提升整体系统效率。**另一方面，**UCM 实现了一系列创新的推理加速算法：自适应全局 Prefix Cache 提供任意位置、任意介质、任意组合的前缀命中能力，可实现 KV Cache 在单机 HBM、单机 DRAM、跨机 DRAM、以及 SSD 池化存储等不同类型、范围中的精准查找和匹配，同时支持公共前缀、历史对话和 RAG 知识块多种拼接组合场景的复用，以提升推理 TTFT 指标；全流程稀疏加速算法提供 Prefill 阶段的超长 KV 分片卸载和增量稀疏，以及 Decode 阶段的动态稀疏，支持覆盖全流程的稀疏加速套件，可倍数级提升长序列推理吞吐及时延；后缀检索预测加速算法将行业的私域数据和用户习惯等隐私数据构建为

Token 级后缀索引，突破推理自回归瓶颈，实现单次输出多词，效果优于传统 MTP。**此外，**UCM 北向支持多样化的推理引擎，南向可接入多样化的存储系统。

该方案已在金融行业典型推理应用场景得到效果验证。在舆情分析系统中，将长文本分类知识库提前预热至 KV Cache Pool 中存储，避免每次推理时重复计算，将推理时延由 10 分钟降低至 10 秒，分类准确率提升至 80%以上，实现客户之声高效精准分析。在会议纪要生成等长文档处理场景中，采用 KV Cache 稀疏去噪技术，将原始上下文进行压缩，突破上下文窗口限制，满足客户对会议纪要准确性与完整性的要求。

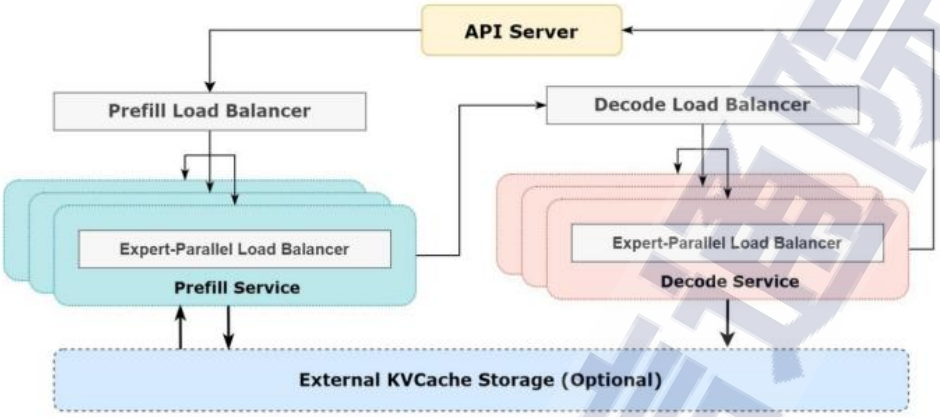
4.协同优化趋势二：结合 MoE 模型特性的架构优化

当前，以 DeepSeek（深度求索）、MegaScale-Infer（字节跳动）、Step-3（阶跃星辰）等为代表的工业级方案持续推出，以上方案基于 PD 分离的系统架构，同时针对 MoE 模型结构在系统层面实现协同优化。其中，MegaScale-Infer（字节跳动）与 Step-3（阶跃星辰）在 PD 分离基础上，针对解码阶段 MoE 层注意力模块与前馈网络模块的计算差异，进一步细化提出了 AF 分离架构，推动推理系统在 MoE 架构下的范式演进。

1) DeepSeek 架构方案

2025 年 2 月，我国深度求索推出的 DeepSeek V3/R1 技术方案。该方案实现超大规模模型高效训推、高阶推理能力等突破，凭借模型架构创新与全链路工程优化，在性能、成本与实用性间实现精准平衡，

一经推出便引发国内外广泛关注，迅速渗透产业应用，成为大模型推理落地的主流选择。



来源：https://github.com/deepseek-ai/open-infra-index/blob/main/202502OpenSourceWeek

图 17 Deepseek 推理系统架构图

其推理阶段方案同样基于 PD 分离架构，核心目标为在保证在线 SLO 的同时最大化吞吐量，该方案围绕大规模 DeepSeekMoE 模型特性进行了深度与细致的工程优化。**并行策略层面，**根据预填充、解码两阶段的计算特性，分别采用不同规模专家并行进行策略组合，以实现预填充阶段充分提升计算效率，解码阶段最大化并行度。**负载均衡层面，**通过在线监测识别热点专家并创建副本，以及冗余专家部署配合动态路由方案，解决推理阶段专家负载不均导致的“木桶效应”。**通信优化层面，**提出双微批次重叠执行[46]实现计算-通信重叠。在预填充阶段将一个微批次的“attention+MoE”与另一个微批次的“dispatch+combine”重叠执行，使通信开销被计算开销覆盖；在解码阶段将一个微批次的“attention”与另一个微批次的“dispatch+MoE+combine”重叠执行，减少解码阶段注意力计算的高耗

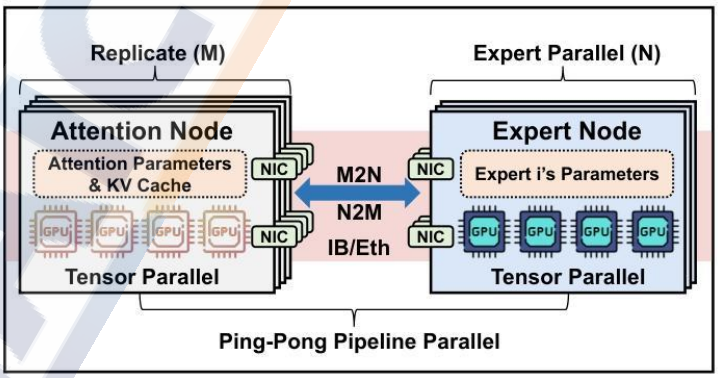
[46] https://github.com/deepseek-ai/profile-data

时占比。**存储优化层面，**通过 3FS 高性能并行文件存储系统和 KV Cache 多级存储架构，将服务成本降低了 90%，并显著降低推理时延。

该方案已实现大规模生产级落地，基于 H800 GPU 部署的推理服务，平均每台预填充吞吐达 73.7k tokens/s（含缓存命中），解码输出吞吐约 14.8k tokens/s。成本层面，通过架构优化与缓存复用，理论成本利润率可高达 545%。其进一步围绕 MoE 架构模型深度优化的技术路径，为大模型推理方案提供了又一成熟范式。

2）MegaScale-Infer 架构方案

2025 年 7 月，字节跳动与北京大学联合发布 MegaScale-Infer 推理系统，核心突破在于通过“A/F 模块解耦+乒乓流水线并行+高性能 M2N 通信库”，解决 MoE 模型在推理过程中，尤其在解码阶段由于 FFN 与 Attention 模块计算需求不同引发的 GPU 利用率低问题，以及数据在专家间分发调度引发的通信延迟问题。该优化方案普遍适用于 MoE 架构模型，已在字节跳动内部大规模部署，为超大规模 MoE 模型的工业化落地提供了系统级范式。



来源：arXiv:2504.02263v4

图 18 MegaScale-Infer 运行时实例架构图

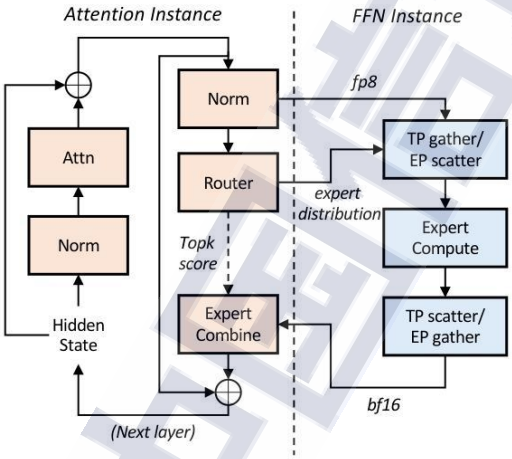
该方案专为 MoE 模型设计，创新性重构了 MoE 推理的模块组织与资源调度模式。**一是，**提出离散化专家并行（Disaggregated Expert Parallelism）架构（即 AF 分离），实现 MoE 模型层内 Attention 模块与 FFN 模块的物理解耦。同时，进一步为两类节点设计差异化的并行策略，以匹配其计算与内存特性。**二是，**提出乒乓流水线并行（Ping-Pong Pipeline Parallelism），通过拆分批次与流水线调度，实现计算-通信重叠，消除 A/F 两类节点间因等待计算/通信引发的频繁闲置，以及跨界点通信带来的时延消耗。**三是，**针对解耦架构下注意力节点（M 个发送方）与专家节点（N 个接收方）形成的“M2N 通信模式”，提出了定制化的高性能 M2N 通信库。通过消除冗余操作、优化流量调度与保障硬件直连，为解耦架构下的 MoE 推理系统提供低延迟、高可靠的通信支撑[47]。

MegaScale-Infer 方案整体更侧重系统层创新，以 A/F 模块解耦打破 MoE 模型层困境与硬件资源限制，支持两类节点的异构硬件配置，适配所有大规模 MoE 架构模型，适用于云原生 AI 服务、弹性推理平台、大规模 API 服务等对吞吐量与成本敏感的场景。该方案已在字节跳动内部实现生产级部署，支撑 Mixtral 8x22B、DBRX 等主流 MoE 模型的线上服务。真实负载下，单 GPU 解码吞吐量较 vLLM 等基线方案最高提升 7.11 倍，异构集群中单位成本吞吐量较 vLLM 提升 224%。**其 A/F 模块解耦与异构部署理念，与 PD 分离架构形成互补，共同推动大模型推理系统的架构革新。**

[47] MegaScale-Infer: Serving Mixture-of-Experts at Scale with Disaggregated Expert Parallelism https://arxiv.org/pdf/2504.02263

3) Step-3 架构方案

2025 年 7 月，我国阶跃星辰推出基于 MoE 架构的 Step-3 自研多模态大模型（VLM），并结合模型进一步提出高效推理架构方案。该方案同样加入了 AF 分离元素，实现“AF 分离+多矩阵分解注意力（MFA）+StepMesh 通信库”的模型-系统协同设计，该优化方案适用于 MoE 架构模型，且已实现对于主流国产芯片的深度适配。



来源：arXiv:2401.09670v3

图 19 AF 分离模块架构图

该推理系统方案深度结合大规模 MoE 模型特性。**一是，**提出多矩阵分解注意力（MFA）的模型结构创新，可在保持与 DeepSeek-V3 相当的高注意力表达能力的同时，显著降低 KV 缓存大小和计算量。**二是，**提出注意-前馈网络解耦（AFD, Attention-FFN Disaggregation）架构，各自采用最优并行策略和硬件配置。**三是，**提出 StepMesh 通信库以更好支持 AFD 的模块解耦和异构部署。

该方案已通过生产级场景验证，在 Hopper GPU 集群上实现 4039 tokens/GPU/s 的推理吞吐量（在 50ms 的 SLA 约束下），是 DeepSeek-

V3（2324 tokens/GPU/s）的 1.73 倍；在国产芯片部署时，推理效率较主流方案提升显著，充分适配国产化算力生态。成本控制方面，其 8K 上下文每百万令牌推理成本低至 0.055 美元，为 DeepSeek-V3（0.068 美元）的 80%，32K 长上下文场景成本低至 0.129 美元，优势进一步扩大，仅为 DeepSeek-V3（0.211 美元）的 61%。**该方案的 AF 分离与异构部署思想与 MegaScale-Infer 异曲同工，共同推动了在 MoE 模型场景下推理系统范式的巩固与迭代**[48]。

五、大模型推理优化典型案例

（一）金融领域

1 案例名称：

金融领域大模型高性能推理加速方案

2 案例实施单位：

某金融清算机构，华为技术有限公司

3 案例介绍：

在国家大力推动人工智能产业发展的战略背景下，某金融清算机构，启动了在“AI+支付”领域的战略性探索。该项目总体投资达数亿元，建设周期为三年，旨在构建一个覆盖“一个算力底座、两大平台（模型平台与数据平台）、六大中心”的先进技术体系。项目的核心目标是在金融普惠、风控合规、消费拉动及质效提升四大关键领域，成功落地超过十个示范性 AI 应用场景，以引领金融行业大模型的规模化应用与创新。

[48] Step-3 is Large yet Affordable: Model-system Co-design for Cost-effective Decoding https://arxiv.org/pdf/2507.19427

1) 整体方案:

在项目落地过程中，金融机构在核心的智能客服、舆情分析、会议纪要生成等场景中，遭遇了两个突出的技术瓶颈:

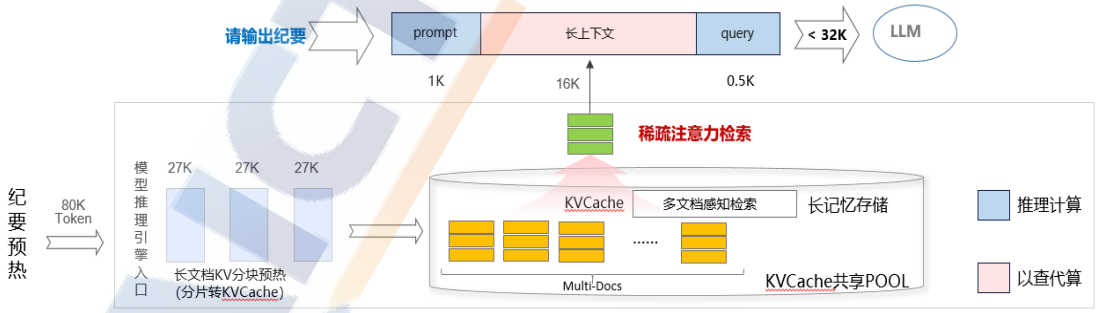
“推不动”：在处理坐席记录、调研报告、会议文本等长文档时，文本序列长度经常超过模型 32K 的上下文窗口限制，导致推理任务无法执行。

“推得慢”：在企业级智能问答等高并发场景下，随着输入序列变长，系统响应速度急剧下降，30 路并发时 TTFT 超过 30 秒，严重影响了业务效率与用户体验。

为解决这些挑战，金融架构引入了华为的 AI 推理加速与存储优化一体化方案，并针对不同业务场景进行了深度适配。

2) 创新情况:

本案例的创新性体现在通过多项自研技术，实现了大模型在金融场景下的高性能、低成本落地:



来源：某金融清算机构、华为技术有限公司

图 20 金融清算场景会议纪要案例方案示意图

基于 AI 存储的 KV Cache 管理技术：在舆情分析系统中，创新性的将长文本分类知识库以 60KB 的 KV Cache 形式提前预热并持久

化在 OceanStor A 系列存储池中。该技术避免了每次推理时对固定知识的重复计算，从根本上降低了长文本处理的时延。

KV Cache 动态稀疏与去噪技术：针对会议纪要等超长文本场景，该技术能对原始上下文进行智能压缩，在不损失关键信息的前提下，将 80K 的上下文负载有效压缩至 16K 进行加载，从而突破了模型固有的上下文窗口限制。

融合“以查代算”的多轮对话记忆机制：在智能问答场景中，系统将“多轮对话历史记忆”与高命中率的缓存查询机制相结合。这不仅有效扩展了有效的对话上下文长度，还通过直接调用缓存结果替代重复计算，显著提升了系统吞吐与响应速度。

3) 应用实效:

通过上述创新技术的精准落地，金融机构在多个业务场景中取得了显著的成效:

性能大幅提升：在舆情分析场景中，端到端推理时延从超过 15 分钟缩短至近 90%，实现了近乎实时的分析能力。在智能问答场景中，8 卡部署下 30 并发 TTFT 降至 9 秒，降低 66%，同时集群吞吐量提升 43%。

处理能力突破：成功突破模型上下文窗口限制，能够一次性处理完整的超长会议内容与多轮对话历史，解决了“推不动”的核心痛点。

输出质量保障：在实现高性能的同时，确保了输出内容的高质量。以会议纪要生成为例，长会议纪要可直接推理，推理结果准确度大幅提升，满足金融业务对准确性与完整性的高标准要求。

用户体验飞跃：智能助手因具备真正的多轮对话记忆和理解能力，实现了从“答非所问的伪助手”到“精准连贯的真助手”的体验跃升，为“AI+支付”的规模化应用奠定了坚实的技术基础。

（二）运营商领域

1 案例名称：

九天人工智能平台

2 案例实施单位：

中移九天人工智能科技(北京)有限公司(九天人工智能研究院)

3 案例介绍：

九天人工智能平台针对大模型推理业务规模化扩张中的核心痛点，构建全栈优化体系，目前已在能源、航空、农业、物流交通等关键行业的大模型生产级业务中实现深度落地。

随着九天人工智能平台用户规模的持续扩大与业务场景的不断延伸，大模型推理业务逐渐暴露出一系列行业共性难题，严重制约了产业大模型的落地效率与应用价值：一是模型训练与推理链路不连贯，不同参数规模模型推理适配流程重复繁琐、部署周期长；二是模型推理业务的资源利用率有待提升，采用 PD 混部的方式使得平台难以在保证稳定时延的前提下承载更多并发会话和更长的上下文长度；三是传统推理框架在底层逐层执行独立算子的过程中，频繁显存读写占据了大量带宽，叠加整体优化手段不足，严重制约了模型推理效率和效能的规模化提升。为支撑大模型训练与推理的一体化演进，进一步兼顾稠密模型在性能、成本与工程可用性的平衡，九天人工智能平台从

架构层、系统层与算子层三个层次制定优化策略。

1）整体方案:

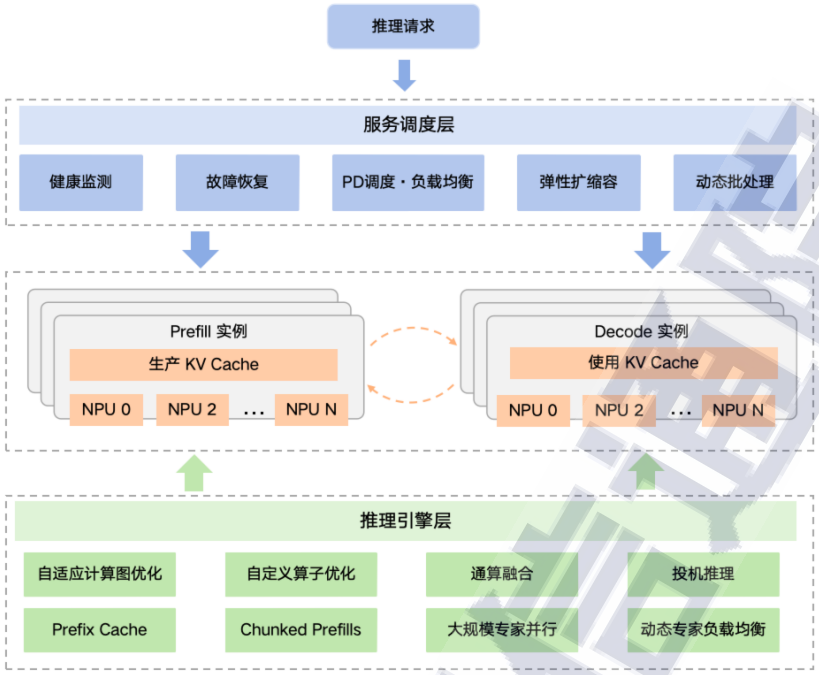
在整体架构上，采用训推一体框架打通训练-推理全链路，统一计算图与并行策略，实现模型的快速迁移与高效运行；在系统层面，引入 PD 分离部署，对不同阶段的计算与存储负载进行精细拆分和调度，提升算力利用率并降低大规模服务的硬件投入；在算子层面，围绕矩阵乘、Attention 与位置编码等关键算子开展融合与深度优化，释放长上下文和高并发场景下的大模型推理性能。

(1) **训推一体架构：**依托训推一体架构，大模型在训练与推理阶段共享同一套计算图和并行策略，兼顾大吞吐训练与低延迟推理。通过增量推理、Flash Attention、Paged Attention 等融合算子的统一加速技术，在保证数值一致性的同时显著提升长上下文推理性能（支持最大 128k 序列长度）。统一训练与推理并行策略接口，对推理加速能力的模块化封装，无需导出、切分和转换模型，实现从训练到高性能推理的平滑迁移，整体部署周期缩短至天级。平台层面，围绕训推一体框架构建统一的算力资源池，整合训练集群与推理集群，并通过统一的调度策略实现跨业务协同编排，低成本实现“昼推夜训”、“闲时训练”等弹性调度策略，减少资源空转与割裂建设带来的成本浪费，提升训推集群的整体算力利用率。

(2) **PD 分离：**PD 分离架构通过将计算密集型的 Prefill 阶段与存储密集型的 Decoding 阶段解耦并分布式部署，实现了资源与阶段特性的精细匹配。同时基于以 KV Cache 为中心的分布式存储架构实

现全局多级 KV Cache 管理，显著降低了因资源竞争导致的 OOM 风险与时延抖动。并且在资源动态调度的场景中，通过自动扩展机制，实时监控业务负载情况，动态调整 P/D 集群的规模。相比混部的推理方式，某稠密模型在引入 PD 分离架构后单卡吞吐量提升 1 倍以上，同时在生产场景中显著降低了大规模服务的硬件成本。

(3) **算子融合：**引入流水线并行与 Swizzle 数据重排策略，通过“以算代搬”掩蔽数据搬运开销、提升缓存命中率，从而挖掘矩阵乘算子的极致性能；依托 Cube/Vector 并行能力，将矩阵乘后的后处理计算与相邻 Vector 类算子进行跨界融合，减少内核调用与中间结果回写，降低计算耗时和数据搬运带来的内存开销；结合芯片特性设计 Tiled-Attention 算法，充分利用片上缓存高带宽，通过构建多级流水与重组 query 计算路径，实现 Attention 相关算子的深度融合与流水并行最大化；同时，将 RoPE 抽象为可融合的编码算子，结合位置编码特性重构计算逻辑与数据布局，最大化硬件缓冲利用率，完成高性能编码算子的融合实现。



来源：中移九天人工智能科技（北京）有限公司（九天人工智能研究院）

图 21 九天人工智能平台优化方案示意图

未来，平台将持续深化推理优化技术研发，推动更多行业大模型实现高效、稳定、低成本的规模化应用，为产业智能化升级提供更坚实的 AI 底座支撑。

（三）电力领域

1 案例名称：

面向中压配网检修业务的推理优化

2 案例实施单位：

中国电力科学研究院

3 案例介绍：

随着分布式新能源与新型用电负荷接入，配网结构日趋复杂，检修计划制定作为配网运维核心，需综合设备运行状态、用户负荷、气

象条件、保电需求等多维度信息。传统大模型推理在此场景中存在三大突出问题：**一是推理时延过高，难以满足时效性要求。**中压配网检修需在夜间窗口期完成次日计划推理，而传统系统处理全区域数据时单次耗时远超窗口期，导致计划延迟发布，打乱运维节奏并影响用电保障。**二是长上下文处理能力不足，制约推理精度。**检修推理需依托设备全生命周期数据，传统方案受 KV Cache 显存限制，仅能截取近期部分数据，丢失设备老化规律等关键信息，导致故障预判不准，存在漏检或过度检修问题。**三是场景适配性差，无法应对复杂工况。**常规检修、故障抢修、特殊保电等场景需求差异显著，但传统系统采用统一推理策略，难以匹配不同场景的时延、优先级等要求，需人工二次调整，降低运维效率。

1) 整体方案：

本方案以中压配网检修业务特性为核心，从模型、推理引擎、调度层三大关键层面构建全链路优化方案，系统涵盖数据采集、预处理、模型、推理引擎、调度与业务应用多层级，各层级协同联动，实现技术与业务深度融合。

模型层优化：针对传统模型痛点，进行 MoE 架构重构与轻量化改造。按配网检修核心需求拆分多个专家网络，各专家网络聚焦设备评估、故障识别等专项任务，通过门控网络动态激活 2-3 个相关专家网络，降低算力消耗。采用混合精度量化策略，关键层保高精度、非关键层低精度处理，同时量化压缩 KV Cache 并修正误差，减小模型体积与显存占用，支撑长上下文全量数据推理。

推理引擎层优化：围绕长上下文处理与低时延需求突破性能瓶颈。设计 KV Cache 多级存储架构与时序预取机制，解决显存瓶颈并降低 I/O 延迟；通过算子融合、硬件适配、计算复用技术加速计算；采用场景感知批处理策略，结合线路优先级调度，按常规检修、故障抢修、特殊保电场景动态调整参数，平衡效率与准确性。

调度层优化：构建“场景感-策略匹配-资源调度”闭环机制。通过场景识别模型精准判断业务场景，基于预设“场景-推理参数”映射关系自动匹配最优参数。依托分布式推理集群，采用负荷感知动态分配策略，按场景需求分配资源，搭配弹性扩容机制，适配不同场景需求并应对业务波动。

2) 场景适配:

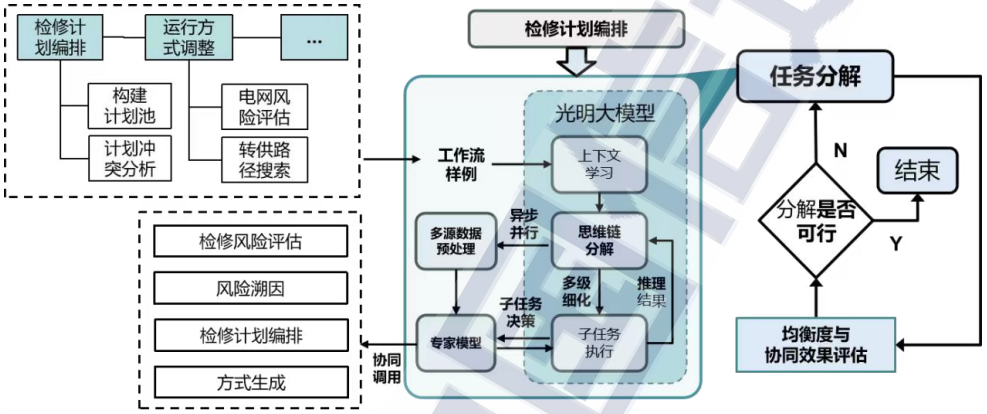
为确保系统契合电力行业安全规范与中压配网检修业务需求，从数据处理、安全合规、业务流程三大维度开展定制化适配，保障系统在实际环境中稳定可靠运行。

多源数据适配：针对配网时序运行数据、结构化台账、非结构化报告与图像等多样数据类型，开发专用预处理插件。按数据特性设计专属逻辑：时序数据补全与异常过滤，结构化数据编码归一化，非结构化数据基于电力百科分词与图像特征提取，统一转化为标准化格式。预处理与推理引擎异步并行，缩短数据处理耗时，提升整体效率。

安全合规适配：严格遵循电力行业等保要求与监控系统安全防护规定。数据传输采用加密算法，防范敏感信息泄露；推理结果添加数字签名，保障未被篡改；系统部署于电力专用安全区域，与外部网络

物理隔离。集成操作日志审计模块，完整记录推理请求相关信息，日志留存满足行业规范，通过合规检测。

业务流程适配：贴合检修“计划生成-审核-下发”全流程，增设针对性功能模块。推理后自动比对设备禁止检修时段，冲突时触发告警；支持运维人员可视化查看调整结果，修改后系统自动重算关联线路影响，确保方案可行性，避免人工调整引发的冲突问题。



来源：中国电力科学研究院

图 22 中压配网检修业务的推理优化方案示意图

（四）司法检察领域

1 案例名称：

检察院“数字检察”项目

2 案例实施单位：

某人民检察院，华为技术有限公司

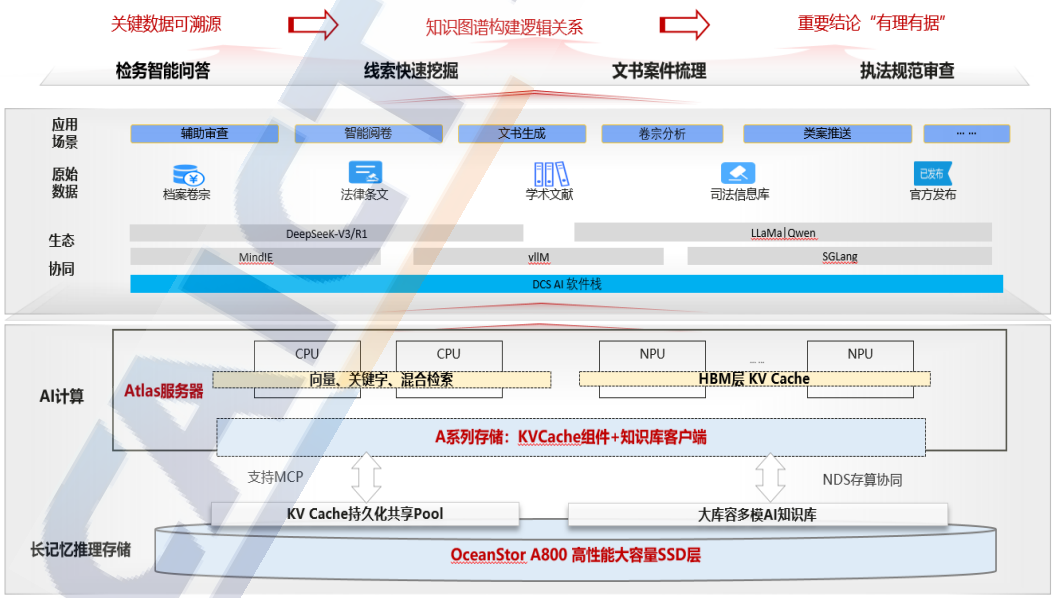
3 案例介绍：

某人民检察院作为智能化建设首批试点单位，致力于打造“数字检察”地标性工程，重点解决“信息孤岛”与“数据烟囱”问题，构建大数据法律监督体系，推动法律监督实现从量到质的提升。面对案多人少、

案情繁杂、知识库庞大复杂、标准极其严格等多重严峻挑战，项目以高性能 AI 存储与昇腾算力为基础，构建覆盖智能问答、案卡填充、卷宗分析等场景的智能化平台，旨在实现检察业务质效的全面革新。

1)　整体方案：

项目整体采用“存算协同”架构，训练集群由 8 台 800T A2 昇腾服务器搭建，推理集群由 8 台同型号服务器与高性能 AI 存储 OceanStor A800 共同组成，并配备通用服务器、存储及数据通信设备，为检察核心数据提供统一底座支撑。方案通过 CCAE 等智能化管控平台实现资源池化管理，借助容器化技术封装 AI 应用，为上层业务系统提供稳定可靠的运行环境。平台针对海量非结构化卷宗信息识别提取、超大规模法律知识检索与实时更新、以及高标准法律文书精准生成等核心业务场景，提供了强有力的技术支撑。实现更低时延与更大吞吐，全面支撑法律监督业务的高效运转。



来源：某人民检察院、华为技术有限公司

图 23　检察院“数字检察”项目系统架构图

2)　创新情况:

“以存助算”架构创新：利用高性能 AI 存储的长记忆内存能力，在推理过程中加载缓存，避免重复计算，显著减轻算力压力，实现毫秒级响应。

长序列数据处理创新：采用 KV Cache 缓存机制，对长卷宗等缓存数据进行分级存储，攻克长文本处理瓶颈，提升法律文书生成质量。

知识库动态更新机制：引入 RAG 技术构建法律知识库，实时更新法律条文与判例，有效约束内容生成，消除模型“幻觉”，推动检察业务从“经验决策”向“数据驱动”转型。

3)　应用实效:

通过该方案的实施，系统在推理阶段 TTFT 降低 40%，吞吐量提升 5 倍，大幅提升了智能问答、案卡回填与卷宗分析等场景的处理效率。法律文书生成质量得到显著改善，数据处理与知识检索的精准度进一步增强，为检察院构建高效、智能的法律监督体系提供了坚实的技术支撑。

（五）农畜领域

1　案例名称:

面向农畜养殖业务的推理优化

2　案例实施单位:

某养殖场，北京百度网讯科技有限公司

3　案例介绍:

某养殖场区每年都会发生不同程度的安全事故。为了降低风险、

提升管理效率，企业希望借助视频监控与智能识别技术，对现场饲养员作业行为进行实时监测。一旦识别出违规操作，系统能够自动推送告警给现场负责人，便于第一时间干预和处置。

场区目前分布着大量摄像头，仅依靠人工值守不仅工作量巨大，且容易出现漏判、误判。同时，随着管理制度不断更新，基于传统“小模型”方式进行模型迭代的周期较长，往往无法及时响应快速变化的业务需求，因此企业开始探索使用大模型进行统一智能识别的方案。

1) 主要场景与存在问题:

- **有限空间作业防护违规**

牲畜环境要求严格，饲养员没有穿防护服或者暴露衣着作业，存在病菌感染风险。

- **有限空间作业违规**

智能养殖操作，包括刷圈、防疫、消毒、加水、上饲料、设备异常等，流程稍有遗留会存在风险。

园区几千名饲养员，每个饲养员负责千头牲畜日常、卫生打扫等。为了监控饲养员操作规范，当前系统采用本地部署的多模态大模型进行视频图像识别，整体场景精度约 60.99%。虽然在一定程度上减少了人工判断，但仍有较多延迟、漏报，无法完全满足业务对高实时性的要求，干预效率低下。

2) 整体方案:

本方案以智能监控饲养员作业业务特性为核心，从模型、推理引擎、调度层三大关键层面构建全链路优化方案，系统涵盖数据采集、

预处理、模型、推理引擎、调度与业务应用多层级，各层级协同联动，实现技术与业务深度融合。大模型将 Prefill 和 Decode 两推理阶段分开处理的技术，通常适用于对时延有严格要求的场景。PD 分离服务部署可以提高 GPU/NPU/XPU 的利用率，尤其是大语言模型，将 Prefill 实例和 Decode 实例分开部署，减少 Prefill 阶段和 Decode 阶段分时复用在时延上造成的互相干扰，实现同时延下吞吐提升。

不同加速卡部署方案

*** GPU PD 分离方案：** Scheduler（单 Pod 副本）+ Prefill + Decode

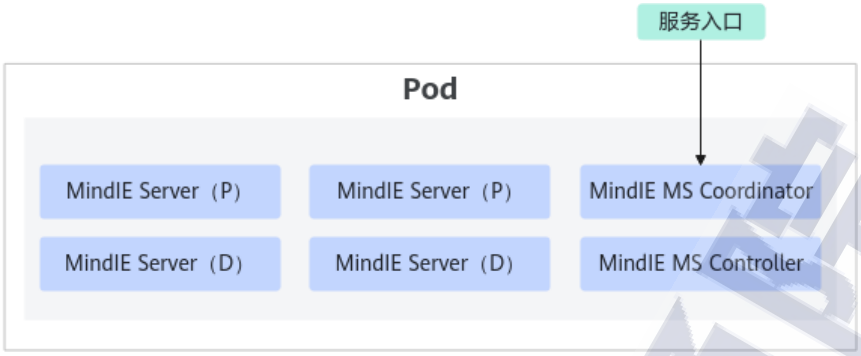
*** XPU PD 分离方案：** Scheduler（单 Pod 副本）+ Prefill + Decode

*** NPU PD 分离方案：** MindIE MS Controller（单 Pod 副本）、MindIE MS Coordinator（单 Pod 副本）以及 MindIE Server（区分 Prefill 和 Decode，由 Mindie 动态调度 P 进程和 D 进程）

* P 实例和 D 实例通过 Headless Serive 传输 KV-Cache，XPU 原始方案中使用 redis 共享 KV-Cache，由推理镜像内部闭环。

以 NPU 为例：

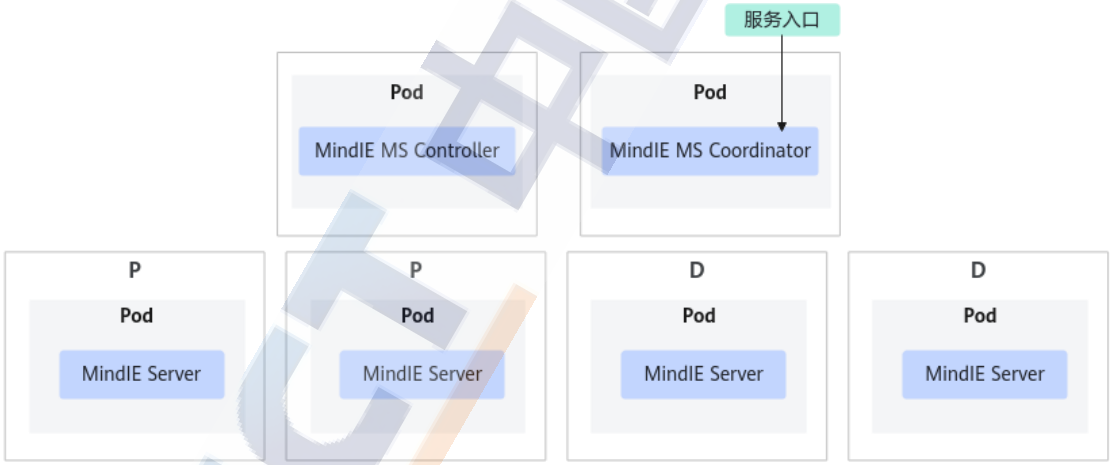
*** 单机 PD 分离服务部署方案：** 通过 K8s 的 Service 开放 PD 集群的推理入口，创建 1 个 K8s 的 Deployment 部署一个 Pod，其中以进程方式分别部署 MindIE MS Controller（单进程副本）、MindIE MS Coordinator（单进程副本）以及 MindIE Server（多进程副本）。



来源：某养殖场、北京百度网讯科技有限公司

图 24 单机 PD 分离方案示意图

* **多机 PD 分离服务部署方案：**通过 K8s 的 Service 为 Coordinator Pod 开放 PD 集群的推理入口，创建 3 个 K8s 的 Deployment 分别部署 MindIE MS Controller（单 Pod 副本）、MindIE MS Coordinator（单 Pod 副本）以及 MindIE Server（多 Pod 副本）。



来源：某养殖场、北京百度网讯科技有限公司

图 25 多机 PD 分离方案示意图

六、展望

大模型推理优化正朝着“协同化、智能化、场景化”的方向深度演进，技术突破与产业需求的深度耦合将重塑推理服务生态。未来，“模型-架构-场景”协同优化将成为核心范式，模型设计将深度融入推理友

好性考量，架构创新将进一步适配不同场景的性能诉求，形成全链路优化闭环。

异构算力与解耦架构将走向精细化协同。以 PD 分离、AF 分离为代表的推理解耦架构，将与擅长计算密集、访存密集等不同负载的各类硬件深度协同，通过阶段分层调度、模块异构部署，实现算力资源与任务特性的精准匹配，在提升吞吐与能效比的同时，压缩单 Token 推理成本。自适应调度与智能化推理将成为主流。通过结合 SLO 的资源配置、并行策略调整及缓存管理，实现负载波动下的自动适配与性能稳定，大幅减少人工干预。

多模态与长序列推理优化将迎来突破，KV Cache 高效管理、跨模态数据协同计算、长序列并行处理等技术创新，将打破当前应用边界，支撑更复杂的智能体交互、超长文档分析等场景落地。同时，性能评估与优化的标准化进程将加速，形成统一的指标体系与测试规范，为产业发展提供有序指引。

整体来看，大模型推理优化将推动 AI 服务从“能用”向“好用、省用”跨越，成为赋能千行百业数字化转型的核心引擎。

编制说明

本研究报告自 2025 年 9 月启动编制，分为前期研究、框架设计、文稿起草、征求意见和修改完善五个阶段。面向大模型推理基础设施的技术提供方和应用方开展了深度访谈和调研等工作。

本报告由中国信息通信研究院人工智能研究所撰写，撰写过程中得到了中国人工智能产业发展联盟、华为技术有限公司、中移九天人工智能科技（北京）有限公司、中国电信股份有限公司研究院、中国电力科学研究院有限公司、北京百度网讯科技有限公司、杭州华电能源工程有限公司、京东科技信息技术有限公司、中电信人工智能科技（北京）有限公司、蚂蚁科技集团股份有限公司、北京大学、北京航空航天大学、之江实验室、济南浪潮数据技术有限公司、山东省科学院高新技术产业（中试）基地、北京矩量无限科技有限公司、北京焱融科技有限公司、是石科技（平湖）有限公司、星环信息科技（上海）股份有限公司等单位的大力支持。

中国信息通信研究院 人工智能研究所

地址：北京市海淀区花园北路 52 号

邮编：100191

电话：010-62301618

传真：010-62301618

网址：www.caict.ac.cn