

BidKV: 显存受限推理的KV缓存回收机制



BidKV: KV Cache Reclamation for Memory-Constrained Inference

陈彦博/Chen Yanbo, 王明琪/Wang Mingqi,
张书豪/Zhang Shuhao

(华中科技大学, 中国 武汉 430074)
(Huazhong University of Science and Technology, Wuhan 430074, China)

DOI: 10.12142/ZTETJ.202604002

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260831.0932.002>

网络出版日期: 2026-08-31

收稿日期: 2026-06-01

摘要: 在线大模型推理中, 键值 (KV) 缓存是制约并发服务的关键显存资源。在显存压力下, 顺序回收或容量贪心回收易抢占高重算代价请求, 导致释放的显存空间被重复预填充占用, 从而显著拉长首Token延迟 (TTFT)。为此, 提出BidKV, 将KV回收建模为存算权衡问题, 依据可释放容量与扰动代价选择单位代价收益最高的请求进行回收。BidKV不改变推理结果, 并复用虚拟大语言模型 (vLLM) 与SGLang的原生抢占路径。ShareGPT实验结果表明, BidKV将vLLM的P95的TTFT由6 769 ms降至631 ms, 300 ms服务等级目标 (SLO) 达成率由68.4%提升至83.2%; 在SGLang上, P95的TTFT较原生最近最少使用 (LRU) 降低94.6%, SLO达成率提升38.8个百分点。在双卡昇腾910B2 (TP=2) 实验中, BidKV在3.8请求/秒负载下实现了631.8 ms的TTFT P95与72.0%的SLO达成率。上述结果验证了BidKV的有效性、跨框架可移植性与多卡适用性, 表明显存受限推理应从容量驱动转向存算协同调度。

关键词: 大语言模型推理; 大语言模型服务; KV缓存; 请求调度; 受害请求选择

Abstract: In online large language model (LLM) inference, the key-value (KV) cache is a critical memory resource that constrains concurrent serving capacity. Under memory pressure, sequential or capacity-greedy eviction policies tend to preempt requests with high recomputation costs, causing the freed memory to be repeatedly occupied by redundant prefill operations and consequently amplifying time-to-first-token (TTFT) latency. To address this issue, BidKV is proposed, which formulates KV cache eviction as a memory-computation trade-off problem. BidKV selects the request with the highest unit-cost benefit for eviction based on its releasable capacity and the corresponding perturbation cost. BidKV does not alter the inference results and reuses the native preemption paths of virtual large language model (vLLM) and SGLang. Experimental results on the ShareGPT dataset demonstrate that BidKV reduces the P95 TTFT latency of vLLM from 6 769 ms to 631 ms and improves the 300 ms service-level objective (SLO) attainment rate from 68.4% to 83.2%. On SGLang, BidKV reduces the P95 TTFT latency by 94.6% compared to the native least-recently-used (LRU) policy and improves the SLO attainment rate by 38.8 percentage points. In a two-card Ascend 910B2 (TP=2) setup, BidKV achieves a P95 TTFT of 631.8 ms and an SLO attainment rate of 72.0% under a workload of 3.8 requests per second. These results validate the effectiveness, cross-framework portability, and multi-GPU applicability of BidKV, demonstrating that memory-constrained inference should shift from capacity-driven to memory-computation co-scheduled strategies.

Keywords: LLM inference; LLM serving; KV cache; request scheduling; victim selection

引用格式: 陈彦博, 王明琪, 张书豪. BidKV: 显存受限推理的KV缓存回收机制 [J]. 中兴通讯技术, 2026, 32(4): 3-8. DOI: 10.12142/ZTETJ.202604002

Citation: Chen Y B, Wang M Q, Zhang S H. BidKV: KV cache reclamation for memory-constrained inference [J]. ZTE technology journal, 2026, 32(4): 3-8. DOI: 10.12142/ZTETJ.202604002

大语言模型 (LLM) 的在线推理需同时满足高吞吐与低交互时延的双重目标。在实际服务中, 请求在提示长度、生成长度及到达时间上存在显著差异, 且在预填充与解码阶段异步推进。随着并发度与上下文长度的增加, 用于保

存历史词元键值向量的键值 (KV) 缓存逐渐成为主要的运行时显存资源^[1]。以Llama-3.1-8B-Instruct为例, 在bf16精度下每个词元的KV缓存空间约为128 KiB, 一个包含4 096个词元的请求将占用约512 MiB显存。在多请求并发场景下, 可用KV容量极易达到上限。

当KV缓存空间不足时, 推理引擎通常会抢占部分正在

基金项目: 新一代人工智能国家科技重大专项 (2025ZD0123804)

运行的请求，释放其占用的KV缓存，待后续通过重计算路径恢复执行。该机制虽不改变最终输出结果，但会丢弃已完成的中间计算。若被抢占的请求接近生成完成，其恢复过程将消耗大量预填充资源，并重新扩张KV缓存，甚至可能进一步触发新一轮的抢占。由此，释放出的显存并不一定真正用于服务等待中的请求，反而可能被重复的预填充操作所消耗，导致首Token延迟（TTFT）升高，服务等级目标（SLO）达成率下降。

因此，KV回收的关键问题在于选择回收哪一个运行请求，而非是否释放显存。对于请求，其可释放的KV容量记为 r_i ，回收代价记为 C_i 。当系统存在 N 个词元的KV空间缺口时，理想的调度目标是在释放总量不低于 N 前提下，使被回收请求集合的总代价尽可能低。这一目标体现了显存受限推理中的存算权衡：单纯追求释放容量可能抢占高重算代价的请求，而仅关注请求顺序则可能遗漏高收益、低代价的回收对象。

现有工作尚未显式求解跨请求的回收选择问题。虚拟大语言模型（vLLM）主要采用后进先出（LIFO）抢占策略^[2]；Orca关注连续批处理优化^[3]；Sarathi-Serve与DistServe分别针对分块预填充与阶段解耦进行改进^[4-5]，均未对不同运行请求的回收代价加以区分。H2O与StreamingLLM则聚焦于请求内部的KV缓存压缩^[6-7]，缺乏跨请求的全局选择机制。

本文提出代价感知KV回收机制BidKV，将可释放容量与抢占扰动编码为效用信号，在活跃请求间选择单位代价收益最高的对象进行回收。主要贡献包括：构建竞价接口与在线效用排序方法；以轻量适配器形式接入vLLM与SGLang^[8]的原生抢占路径；基于真实对话负载验证了BidKV在TTFT与SLO指标上的改善效果。

1 竞价驱动的效用调度框架

1.1 总体架构

BidKV包含竞价表示、竞价生成、效用排序和运行时适配4个模块，分别负责统一接口、估计回收收益与代价、比较候选请求并执行抢占操作。该分层设计将代价估计与对象选择解耦，便于替换不同的估计器，并支持迁移至不同的推理引擎。

图1展示了BidKV效用调度的总体流程。系统包含两个异步过程：后台排序与压力触发回收。后台排序周期性收集运行请求的KV占用、完成进度及历史抢占状态，计算各请求的回收效用，并将优先级写入缓存表。压力触发回收在每个调度周期检测KV利用率；当利用率达到设定阈值时，运

行时适配器根据缓存优先级调整运行队列，使最高效用请求进入框架原生抢占路径；当KV利用率低于阈值时，系统不干预框架的默认调度行为。

1.2 基于最短作业优先的准入优化

BidKV在准入侧采用最短作业优先（SJF）策略，按提示长度由短到长对等待队列进行重排。提示长度在请求到达时即可获取，以此低成本近似预填充开销，减少短请求被长请求阻塞的情况，从而改善TTFT及SLO达成率。

SJF决定等待请求“谁先进入”，效用排序决定KV压力下“谁先退出”：前者缓解长请求阻塞，后者减少低效抢占，二者共同优化准入响应与重算代价。

1.3 竞价信号与扰动估计

每个运行请求对应一个竞价对象 b 。竞价对象包含请求标识、被抢占后可释放的KV词元数 r 、扰动估计 δ ，以及可选的置信度与策略元数据。其中， r 表示回收收益， δ 表示中断该请求可能引入的系统扰动，包括已完成计算的丢弃、请求重新排队及重复抢占等代价。扰动估计越小，说明回收该请求对系统造成的影响越低。

当前实现使用请求的生命周期特征对扰动代价进行估计：

$$\delta = 1 + w_c c + w_p P \quad (1),$$

其中， c 表示相对于请求配置最大生成长度的归一化解码进度， P 表示此前被抢占次数， w_c 与 w_p 分别为完成度权重与重复抢占惩罚权重，默认取值为0.5与0.3。在线阶段无法预知实际输出长度，因此 c 并非真实的完成比例，而是已投入解码计算量的低成本代理。BidKV仅将其作为软惩罚项，并结合可释放KV容量与历史抢占状态进行相对排序；完成度项用于降低对高进度请求的回收倾向，重复抢占项则有助于减少饥饿风险。第3.5节将进一步检验权重的敏感性。

1.4 基于效用的在线选择

BidKV依据下式计算请求竞价效用：

$$U(b) = r / (\delta + \varepsilon) \quad (2),$$

其中， $\varepsilon = 10^{-3}$ 用于避免分母为零。该效用表示单位估计扰动代价所能释放的KV容量。效用越高，说明该请求能够以较低代价释放较多的KV空间，因此更适合作为优先回收对象。

每轮后台排序按效用 U 由高到低生成优先级序列；KV压力发生时，依次回收高效用请求，直至满足容量需求。运

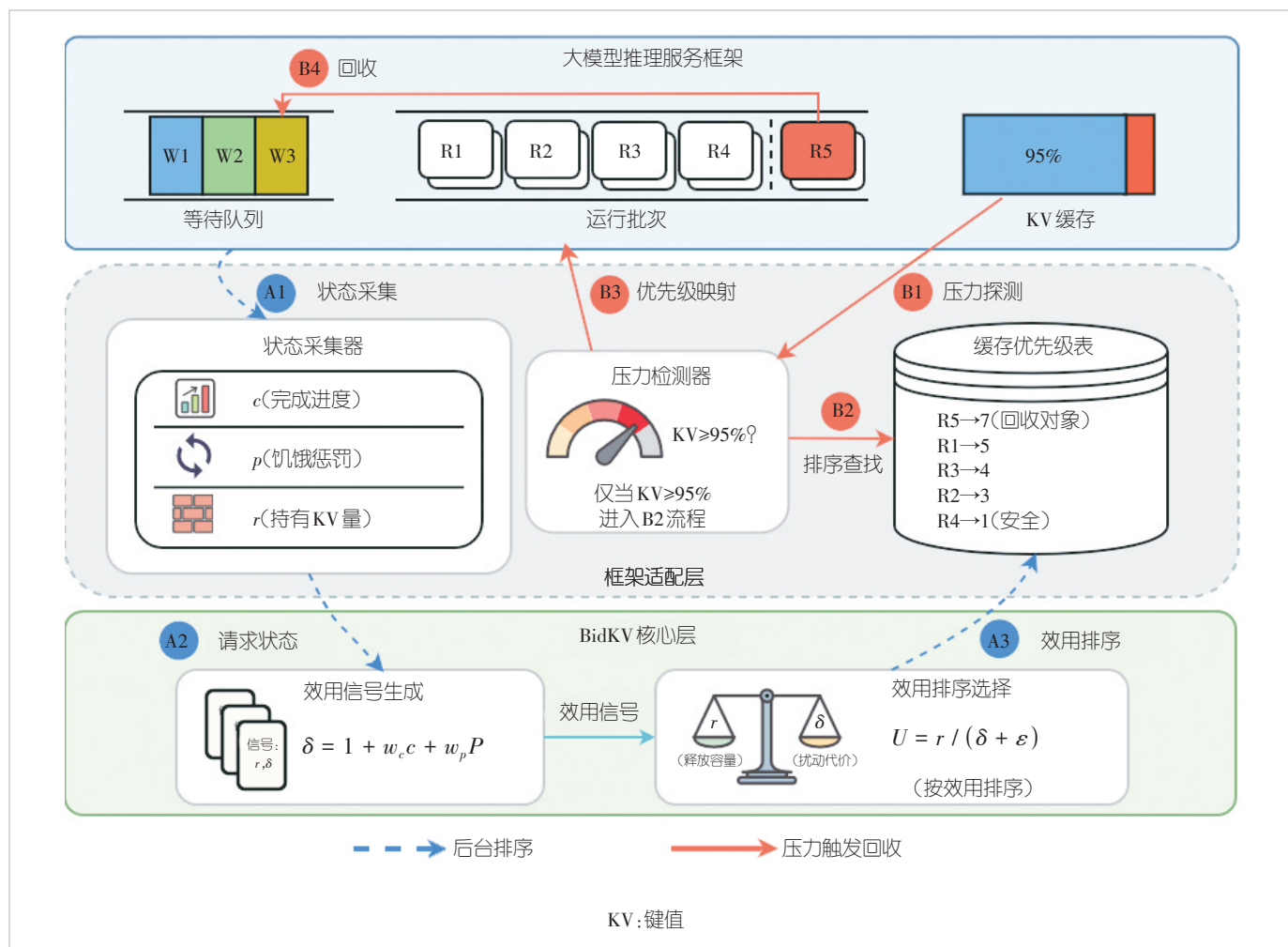


图1 BidKV效用调度总体流程

行请求数为 B 时, 排序复杂度为 $O(B \log B)$, 通常低于一次解码步骤的开销。该排序同时利用 KV 占用与生命周期状态, 优先选择释放容量大、重算代价低的请求。

1.5 运行时适配接口

轻量运行时适配器负责读取 KV 容量、拦截压力事件、调用原生抢占、更新请求状态, 并在完成后清理竞价信息。

BidKV 仅决定“回收哪个请求”, KV 释放、重新排队与恢复执行仍由原框架完成, 因此不改变生成结果或底层内存路径。在 vLLM 中, BidKV 通过队列调整引导原生抢占; 在 SGLang 中, 则以私有后缀估计可释放容量。

2 跨框架轻量化适配机制

BidKV 以独立 Python 包的形式非侵入式接入推理引擎, 通过统一注册接口支持评分策略的替换, 并复用框架原生的

KV 管理与恢复路径。

在 vLLM 0.17.1 中, BidKV 通过插件系统在每次原生调度前执行回调。适配器根据缓存的效用排序调整运行队列, 使目标请求位于 vLLM LIFO 抢占路径的末端, 从而引导原生抢占机制选择最高效用的请求。实际的 KV 块释放、请求重新排队及恢复执行仍由 vLLM 完成, BidKV 不修改 KV 缓存协调器或内存块池。为避免主动抢占引入额外的重算开销, BidKV 仅在 KV 利用率超过 95% 且平均提示长度不高于 500 词元时启用运行队列重排; 低于该条件时, 系统保持框架的默认调度行为。请求完成后, 与该请求相关的效用对象及历史状态将被及时清理。

与 vLLM 不同, SGLang 的适配需要处理其独特的 KV 组织方式。SGLang 采用 RadixAttention 管理 KV 缓存, 允许多个请求共享前缀树中的 KV 节点。在此机制下, 抢占某个请求并不会释放其共享前缀, 仅能释放该请求独有的后缀部分;

共享前缀仍保留于缓存中，且在请求恢复时无须重新计算。因此，若直接以请求的总KV长度作为可释放容量，调度器将高估包含大量共享前缀的请求的回收收益。

为适配这一特性，SGLang适配器将效用公式中的可释放容量 r 替换为请求私有KV词元数 r_{priv} 。在每轮的调度中，适配器通过前缀树节点的引用计数识别共享KV的位置，统计各运行请求独有后缀的长度，并以此来估计真实可释放容量以及相应的重算影响。除该框架相关的调整外，SGLang与vLLM共享相同的效用生成以及效用排序逻辑。该实现不修改SGLang的核心执行与内存管理代码，表明代价感知KV回收可通过少量适配迁移至不同KV组织方式的推理框架中。

3 性能评估与可移植性验证

3.1 实验设置

单卡实验使用1张NVIDIA RTX A6000（48 GiB，CUDA 12.5）与bf16精度的Llama-3.1-8B-Instruct^[9]模型，主要平台为vLLM 0.17.1，KV容量固定为9 600词元。测试负载由从ShareGPT数据集中采样的1 000个混合长度请求构成^[10]，请求按泊松过程到达；每种配置运行3次，并报告测量结果的平均值。多卡补充实验使用2张昇腾910B2，配置张量并行度为TP=2，请求速率为3.8请求/秒（req/s）。

评估指标包括吞吐率、300 ms SLO达成率、TTFT的95分位值，以及每输出词元时间（TPOT）的95分位值。对比方法包括默认抢占策略（PE）、短作业优先抢占策略（PE-SJF）、静态随机回收（Static-Random）、最大容量优先回收（Largest-First）以及本文方法（BidKV）。

3.2 vLLM中等负载性能

表1给出了请求到达率为3.8 req/s时的实验结果。在该负载下，系统持续承受KV压力，但尚未完全饱和，因此能够较为清晰地反映不同回收对象选择策略之间的性能差异。

与默认抢占策略相比，本文方法将TTFT P95从6 769 ms降至631 ms，降幅超过90%，并将300 ms SLO达成率从

68.4%提升至83.2%，绝对值提升了14.8个百分点。PE-SJF的结果表明，仅调整等待队列顺序即可显著改善TTFT，但其TPOT P95升至153.5 ms，这说明了准入优化无法单独解决抢占后的重算干扰问题。本文方法则进一步在运行请求间进行效用排序，在保持较高吞吐率的同时，取得了最低的TTFT P95。

静态随机回收策略虽取得了最高的吞吐率与最高的SLO达成率，但其TTFT P95达到1 216 ms，约为本文方法的两倍，这说明积极抢占虽能提升整体处理速度，却可能加重部分请求的尾部等待。最大容量优先回收策略的TTFT P95为677 ms，但其SLO达成率比本文方法低3.3个百分点。原因在于，KV占用最大的请求通常已执行较长时间，回收后需付出较高的重算代价。本文方法同时兼顾容量收益与扰动代价，在吞吐率仅略低于默认策略的情况下，显著改善了面向用户的首Token响应性能。

3.3 SGLang可移植性验证

为验证该方法能否迁移至不同的推理框架，本文在相同混合负载与3.8 req/s到达率的条件下，将BidKV接入SGLang的RadixAttention后端，并与原生SGLang及随机回收适配器进行比较。

BidKV将SGLang的SLO达成率由54.4%提升至93.2%，TTFT P95由6 207 ms降至337 ms；与随机回收策略相比，亦取得了更高的SLO达成率、更低的TTFT及更高的吞吐率。表2为SGLang可移植性实验结果。上述结果表明，采用私有KV长度作为可释放容量估计，可有效避免因共享前缀导致的回收收益高估，且所设计的效用接口能够通过轻量适配

表2 SGLang可移植性实验结果

策略	300 ms SLO达成率/%	首Token延迟 P95/ms	吞吐率/(req·s ⁻¹)	单Token输出时间 P95/ms
原生方法	54.4	6 207	3.01	58.8
随机回收	92.3	386	3.69	70.1
本文方法	93.2	337	3.72	70.9

SLO:服务等级目标

表1 vLLM中等负载下的性能对比(请求速率为3.8 req/s)

策略	300 ms SLO达成率/%	首Token延迟 P95/ms	吞吐率/(req·s ⁻¹)	单Token输出时间 P95/ms
PE	68.4	6 769	3.48	97.0
PE-SJF	79.6	666	3.06	153.5
Static-Random	84.1	1 216	3.57	90.9
Largest-First	79.9	677	3.28	112.9
BidKV	83.2	631	3.42	107.2

PE:抢占策略 PE-SJF:短作业优先抢占策略 SLO:服务等级目标

迁移至不同推理框架。

3.4 双卡TP并行验证

为贴近多卡部署场景，本文在2张昇腾910B2上配置张量并行度（TP=2），并在3.8 req/s的请求速率下，比较了默认PE、PE-SJF、Static-Random、Largest-First以及本文方法（BidKV）的性能表现。

表3显示，在3.8 req/s的负载下，BidKV的TTFT P95为631.8 ms，SLO达成率为72.00%，吞吐率为1.806 req/s。与PE相比，BidKV的TTFT P95降低了94.2%，SLO达成率提升了25.20个百分点；与Static-Random相比，TTFT P95降低了70.1%，SLO达成率提升了7.78个百分点。PE-SJF的SLO达成率比BidKV高2.35个百分点，但BidKV的TTFT P95低27.4%；Largest-First的TTFT P95比BidKV低14.6 ms，但BidKV的SLO达成率高出5.70个百分点。上述结果表明，BidKV并非追求单项指标最优，而是在双卡TP并行场景下实现了尾部延迟与SLO达成率之间更好的平衡。

3.5 超参数敏感性分析

在与表1相同的负载条件下（3.8 req/s）进行单因素扫

表3 双卡昇腾910B2上TP=2的性能对比(请求速率为3.8 req/s)

策略	300 ms SLO达成率/%	首Token延迟P95/ms	吞吐率/(req·s ⁻¹)	单Token输出时间P95/ms
PE	46.80	10 945.3	1.768	208.4
PE-SJF	74.35	869.7	1.848	199.4
Static-Random	64.22	2 113.2	1.800	196.1
Largest-First	66.30	617.2	1.796	198.5
BidKV	72.00	631.8	1.806	198.8

PE: 抢占策略 PE-SJF: 短作业优先抢占策略 SLO: 服务等级目标

描实验，其余参数固定为默认值： w_c 分别取0.25、0.5、1.0和2.0， w_p 分别取0.1、0.3、0.6和1.0；各配置独立运行。

表4结果显示，吞吐率为3.34~3.43 req/s，SLO达成率为82.3%~84.2%，TTFT P95为600~659 ms，TPOT P95为100.7~112.0 ms。所有配置均显著优于默认抢占策略（SLO达成率68.4%，TTFT P95为6 769 ms）。

在较宽的扫描参数范围内，系统性能保持稳定，这表明基于相对效用的排序机制不依赖于某一组特定的最优权重。默认值 $w_c=0.5$ 和 $w_p=0.3$ 位于扫描区间的内部，能够在保护高进度请求与防止重复抢占之间取得良好平衡。该结果验证的是权重的鲁棒性，并非直接衡量模型对实际终止长度预测误差的容忍度。

4 结束语

本文中我们提出BidKV，以可释放KV容量与扰动代价构造效用函数，在跨请求范围内选择高收益、低代价的回收对象，并复用推理引擎的原生抢占与恢复路径。实验结果表明，BidKV能够显著降低vLLM与SGLang的尾部TTFT，提升SLO达成率，并保持具有竞争力的吞吐率；双卡TP=2的进一步实验验证了其在多卡场景下的适用性。当前采用的归一化解码进度无法精确反映因停止词元导致的提前终止，未来可结合输出长度预测或终止概率估计进行改进。目前，BidKV已集成至华中科技大学vLLM-HUST国产推理引擎工程链路，可为自主可控、低时延、高并发的推理服务提供轻量级调度支撑。

参考文献

- [1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [PP/OL]. arXiv(2023-08-02) [2026-06-16]. <http://arxiv.org/abs/1706.03762>

表4 BidKV超参数敏感性实验(请求速率为3.8 req/s)

参数配置	w_c	w_p	吞吐率/(req·s ⁻¹)	300 ms SLO达成率/%	首Token延迟P95/ms	单Token输出时间P95/ms
默认配置	0.5	0.3	3.39	82.3	633	109.6
$w_c=0.25$	0.25	0.3	3.34	82.8	631	112.0
$w_c=1.0$	1.0	0.3	3.43	84.2	600	100.7
$w_c=2.0$	2.0	0.3	3.35	83.1	659	108.2
$w_p=0.1$	0.5	0.1	3.40	83.5	636	104.3
$w_p=0.6$	0.5	0.6	3.36	83.2	654	105.4
$w_p=1.0$	0.5	1.0	3.42	83.9	640	102.4

注: 默认配置为 $w_c=0.5, w_p=0.3$; 各配置独立运行
SLO: 服务等级目标

- [2] Kwon W, Li Z, Zhuang S, et al. Efficient memory management for large language model serving with PagedAttention [PP/OL]. arXiv (2023-09-12)[2026-06-12]. <https://arxiv.org/abs/2309.06180>
- [3] Yu G I, Jeong J S, Kim G W, et al. Orca: a distributed serving system for Transformer-based generative models [C]// Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI '22). Carlsbad: USENIX Association, 2022: 521-538.
- [4] Agrawal A, Kedia N, Panwar A, et al. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve [PP/OL]. arXiv (2024-06-17)[2026-06-15]. <https://arxiv.org/abs/2403.02310>
- [5] Zhong Y, Liu S, Chen J, et al. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving [PP/OL]. arXiv(2024-06-06) [2026-06-15]. <https://arxiv.org/abs/2401.09670>
- [6] Zhang Z, Sheng Y, Zhou T, et al. H2O: heavy-hitter oracle for efficient generative inference of large language models [PP/OL]. arXiv(2023-12-18) [2026-06-14]. <https://arxiv.org/abs/2306.14048>
- [7] Xiao G, Tian Y, Chen B, et al. Efficient streaming language models with attention sinks [PP/OL]. arXiv(2024-04-07) [2026-06-13]. <https://arxiv.org/abs/2309.17453>
- [8] Zheng L, Yin L, Xie Z, et al. SGLang: efficient execution of structured language model programs[PP/OL]. arXiv(2024-06-06) [2026-06-13]. <https://arxiv.org/abs/2312.07104>
- [9] Dubey A, Jauhri A, Pandey A, et al. The Llama 3 herd of models [PP/OL]. arXiv(2024-11-23) [2026-06-10]. <https://arxiv.org/abs/2407.21783>
- [10] ShareGPT Team. ShareGPT: share your ChatGPT conversations [EB/OL]. [2026-06-10]. <https://sharegpt.com>

作者简介



陈彦博，华中科技大学在读硕士研究生；主要研究方向为大模型推理系统优化。



王明琪，华中科技大学在读硕士研究生；主要研究方向为大模型推理系统优化。



张书豪，华中科技大学教授；主要研究方向为大模型推理引擎、分布式推理、KV Cache与内存管理、服务调度、算子与内核优化；主持国家级及企业科研项目10余项；入选国家高层次人才项目；发表论文40余篇。

中兴通讯技术杂志社成功举办 第31次编委会议暨2026年通信热点技术研讨会

2026年8月1日—2日，中兴通讯技术杂志社第31次编委会议暨通信热点技术研讨会在山西大同成功召开。本次盛会汇聚国内外高校、通信运营商、科研院所、中兴通讯等50家单位的130余位专家学者；陆建华、段宝岩、张宏科、张文军、陈勇5位院士，谈振辉教授、蒋林涛总工等资深编委，中兴通讯董事长方榕、CTO兼杂志社总编王喜瑜、高级副总裁王翔等公司领导出席了会议。

中兴通讯董事长方榕致欢迎辞。她表示，中兴通讯聚焦“连接+算力”双轮战略，持续研发投入，取得一系列成绩。在中兴通讯的创新之路上，两本学术期刊成为了联动产业界与学术界的关键桥梁。方榕董事长希望中兴通讯的两本期刊坚守学术初心，发挥期刊学术引领与产学研平台的协同优势，促进科技创新与产业创新深度融合，发挥编委优势，培养创新人才。

中兴通讯CTO、杂志社总编王喜瑜在致辞中指出，中兴通讯将集中资源攻坚智算核心芯片，自主搭建国产化算力底座。杂志社将坚守“科技向善”办刊定位，打造高水平交流平台。

杂志社副主编卢丹作年度工作报告，系统复盘本年度期刊品质提升、行业影响力建设相关工作，介绍中英文双刊斩获的各类行业荣誉，同步发布下一年度重点工作规划。

会上，陆建华院士作特邀主旨学术报告，系统解读中国天地一体算力网络建设与发展，报告内容引发在场专家广泛关注。

本次会议同步开设中兴通讯未来发展高端论坛，参会院士、学界专家、企业高管围绕企业中长期战略布局、底层技术自主创新等核心议题充分交流，输出多项具备落地价值的建设性意见。

通信热点技术研讨环节，张宏科、艾渤、任奎、赵慧玲、向际鹰、敖立等14位海内外知名专家依次作专题分享，围绕人工智能通信、新一代无线信道、AI网络安全等前沿方向，分享最新科研成果与产业落地实践。

作为年度固定学术交流平台，中兴通讯技术杂志社编委会议打通学术界与产业界沟通壁垒，搭建高效、高质量的双向交流通道，持续为中国信息通信产业技术创新与发展注入动能。