Novelty Assessment Report

Paper: DualMap: Enabling Both Cache Affinity and Load Balancing for Distributed LLM Serving
PDF URL: https://openreview.net/pdf?id=zCadrJ32Xn
Venue: ICLR 2026 Conference Submission
Year: 2026
Report Generated: 2026-01-05

Abstract

In large language model (LLM) serving, reusing the key-value (KV) cache of prompts across requests is a key technique for reducing time-to-first-token (TTFT) and lowering serving costs. Cache-affinity scheduling, which co-locates requests with the same prompt prefix to maximize KV cache reuse, often conflicts with load-balancing scheduling, which aims to distribute requests evenly across compute instances. Existing schedulers struggle to reconcile this trade-off, as they operate within a single mapping space, typically applying cache-affinity routing to a subset of requests and load-balanced routing to the rest, without a unified solution to achieve both goals. To overcome this limitation, we propose DualMap, a dual-mapping scheduling strategy for distributed LLM serving that simultaneously enables cache affinity and load balancing. The key idea of DualMap is to map each request to two candidate instances using two independent hash functions based on the request prompt, and then intelligently select the better candidate based on current system states. This design increases the likelihood that requests with shared prefixes are co-located, while evenly dispersing distinct prefixes across the cluster via ``the power of two choices''. To make DualMap robust under dynamic and skewed real-world workloads, we incorporate three techniques: 1) SLO-aware request routing, which prioritizes cache affinity but switches to load-aware scheduling when TTFT exceeds the SLO, enhancing load balance without sacrificing cache reuse; 2) hotspot-aware rebalancing, which dynamically migrates requests from overloaded to underloaded instances, mitigating hotspots and rebalancing the system; 3) lightweight dual-hash-ring scaling, which leverages a dual-hash-ring mapping to support fast and low-overhead instance scaling without costly global remapping. Experiments on real-world workloads show that DualMap improves effective request capacity by up to 2.25$\times$ under the same TTFT SLO constraints, compared with the state-of-the-art work.

Core Task Landscape

This paper addresses: **Request Scheduling for Distributed LLM Serving with KV Cache Reuse**

A total of **40 papers** were analyzed and organized into a taxonomy with **16 categories**.

Taxonomy Overview

The research landscape has been organized into the following main categories:

- **KV Cache Management and Storage Architectures**
- **Request Scheduling and Routing Strategies**
- **Multi-Request and Multi-Turn KV Cache Reuse**
- **Resource Management and System Optimization**
- **Surveys, Benchmarks, and Cross-Cutting Studies**

Complete Taxonomy Tree

- Request Scheduling for Distributed LLM Serving with KV Cache Reuse Survey Taxonomy
- KV Cache Management and Storage Architectures
  - Memory Hierarchy and Offloading Mechanisms (3 papers)
  - [1] Fast State Restoration in LLM Serving with HCache (Shi-Wei Gao, 2025) View paper
  - [2] MemServe: Context Caching for Disaggregated LLM Serving with Elastic Memory Pool (Huang, 2024) View paper
  - [25] Compute or load kv cache? why not both? (Jin Shuowei, 2024) View paper
  - Disaggregated Inference Architectures (5 papers)
  - [7] A Scalable Approach to Distributed Large Language Model Inference (Yihua, 2025) View paper
  - [12] Mooncake: A kvcache-centric disaggregated architecture for llm serving (Li, 2024) View paper
  - [13] FastDecode: High-Throughput GPU-Efficient LLM Serving using Heterogeneous Pipelines (He, 2024) View paper
  - [32] FlowKV: A Disaggregated Inference Framework with Low-Latency KV Cache Transfer and Load-Aware Scheduling (Li Wei-Qing, 2025) View paper
  - [34] ServerlessPD: Fast RDMA-Codesigned Disaggregated Prefill-Decoding for Serverless Inference of Large Language Models (Mingxuan Liu, 2025) View paper
  - GPU-Centric and Accelerator-Optimized KV Storage (3 papers)
  - [10] FlashInfer: Efficient and Customizable Attention Engine for LLM Inference Serving (Ye Zihao, 2025) View paper
  - [11] Serving Large Language Models on Huawei CloudMatrix384 (ZUO PengFei, 2025) View paper
  - [36] TARDIS: A GPU-Centric KV Cache Service for Efficient LLM Inference (Yifan Hu, 2025) View paper
  - Memory Fragmentation and Block Management (2 papers)
  - [6] Layerkv: Optimizing large language model serving with layer-wise kv cache management (Xiong Yi, 2024) View paper
  - [22] FineServe: Precision-Aware KV Slab and Two-Level Scheduling for Heterogeneous Precision LLM Serving (Choi Seungbeom, 2025) View paper
- Request Scheduling and Routing Strategies
  - Cache-Affinity and Load-Balancing Routing ★ (4 papers)
  - [0] DualMap: Enabling Both Cache Affinity and Load Balancing for Distributed LLM Serving (Anon et al., 2026) View paper

- [17] Preble: Efficient Distributed Prompt Scheduling for LLM Serving (Srivatsa, 2024) View paper
- [26] Online Context Caching for Distributed Large Language Models Serving (Bin Gao, 2025) View paper
- [28] BanaServe: Unified KV Cache and Dynamic Module Migration for Balancing Disaggregated LLM Serving in AI Infrastructure (He, 2025) View paper
- Preemptive and Priority-Based Scheduling (2 papers)
- [20] TokenFlow: Responsive LLM Text Streaming Serving under Request Burst via Preemptive Scheduling (Chen Junyi, 2025) View paper
- [21] No Request Left Behind: Tackling Heterogeneity in Long-Context LLM Inference with Medha (Amey Agrawal, 2024) View paper
- SLO-Aware and Adaptive Scheduling (3 papers)
- [3] Optimizing SLO-oriented LLM Serving with PD-Multiplexing (Cui Weihao, 2025) View paper
- [4] WindServe: Efficient Phase-Disaggregated LLM Serving with Stream-based Dynamic Scheduling (JingQi Feng, 2025) View paper
- [9] Apt-Serve: Adaptive Request Scheduling on Hybrid Cache for Scalable LLM Inference Serving (Gao Shihong, 2025) View paper
- Theoretical Models and Online Algorithms (1 papers)
- [19] Online Scheduling for LLM Inference with KV Cache Constraints (Jaillet, 2025) View paper

- Multi-Request and Multi-Turn KV Cache Reuse
  - Multi-Turn Dialogue and Agentic Workloads (3 papers)
  - [5] Continuum: Efficient and Robust Multi-Turn LLM Agent Scheduling with KV Cache Time-to-Live (Li Hanchen, 2025) View paper
  - [14] Accelerating LLM Serving for Multi-turn Dialogues with Efficient Resource Management (Jin-Woo Jeong, 2025) View paper
  - [33] Tokencake: A KV-Cache-centric Serving Framework for LLM-based Multi-Agent Applications (Wu Feiyang, 2025) View paper
  - Prefix and Context Caching (2 papers)
  - [24] MCaM : Efficient LLM Inference with Multi-tier KV Cache Management (Kexin Chu, 2025) View paper
  - [30] KVShare: An LLM Service System with Efficient and Effective Multi-Tenant KV Cache Reuse (Yang Huan, 2025) View paper
  - Cross-Model and Multi-Tenant KV Sharing (1 papers)
  - [27] DroidSpeak: KV Cache Sharing for Cross-LLM Communication and Multi-LLM Serving (Liu, 2024) View paper
- Resource Management and System Optimization
  - Multi-GPU and Distributed Memory Management (2 papers)
  - [8] FastCache: Optimizing Multimodal LLM Serving through Lightweight KV-Cache Compression Framework (Wu Hang, 2025) View paper
  - [29] Mell: Memory-Efficient Large Language Model Serving via Multi-GPU KV Cache Management (QianLi Liu, 2025) View paper
  - Heterogeneous and Hybrid Execution (2 papers)
  - [23] CE-LSLM: Efficient Large-Small Language Model Inference and Communication via Cloud-Edge Collaboration (Zhu Pengyan, 2025) View paper
  - [40] Sim-LLM: Optimizing LLM Inference at the Edge through Inter-Task KV Reuse (R Luo, n.d.) View paper
  - Compression and Quantization Techniques (1 papers)
  - [16] COMET: Towards Practical W4A4KV4 LLMs Serving (Lian Liu, 2025) View paper
  - Microservices and Modular Architectures (2 papers)
  - [15] A System for Microserving of LLMs (Jin, 2024) View paper
  - [37] xLLM Technical Report (Liu Tong-Xuan, 2025) View paper
- Surveys, Benchmarks, and Cross-Cutting Studies (5 papers)
  - [18] A survey on large language model acceleration based on kv cache management (Li, 2024) View paper
  - [31] LLM Inference Scheduling: A Survey of Techniques, Frameworks, and Trade-offs (Morgan Heisler, 2025) View paper
  - [35] Self-Defining Systems (T Anderson, 2025) View paper
  - [38] SPADA: Secure, Performant, and Distributed LLM Inference (K Chu, n.d.) View paper
  - [39] Workshop on Mobility in the Evolving Internet Architecture to be held in conjunction with MobiComï¿½ï¿½ï¿½ï¿½ (SIGMOBILE, n.d.) View paper

Narrative

Core task: request scheduling for distributed LLM serving with KV cache reuse. The field has organized itself around several complementary dimensions. One branch focuses on KV cache management and storage architectures, exploring how to efficiently store and retrieve cached key-value tensors across memory hierarchies and distributed nodes. Another branch examines request scheduling and routing strategies, addressing how to direct incoming queries to servers in ways that maximize cache hits while balancing load. A third branch investigates multi-request and multi-turn KV cache reuse, tackling scenarios where prefixes or conversational context can be shared across users or sessions. Resource management and system optimization studies address broader questions of memory allocation, batching policies, and end-to-end throughput. Finally, surveys and benchmarks provide cross-cutting perspectives on the evolving landscape, as seen in works like KV Cache Survey[18] and Inference Scheduling Survey[31].

Within the scheduling and routing branch, a particularly active line of work balances cache affinity against load distribution. Some systems such as Preble[17] and Online Context Caching[26] prioritize routing requests to instances that already hold relevant cached prefixes, aiming to minimize redundant computation. Others like BanaServe[28] incorporate load-aware heuristics to prevent hotspots when popular prefixes concentrate traffic on a few servers. DualMap[0] sits squarely in this cache-affinity and load-balancing cluster, proposing mechanisms that jointly consider prefix overlap and server utilization when making routing decisions. Compared to Preble[17], which emphasizes prompt-aware scheduling within a single datacenter, DualMap[0] extends the approach to multi-tier or geo-distributed settings where network latency and hierarchical cache placement become additional factors. Meanwhile, BanaServe[28] explores dynamic rebalancing under skewed workloads, a complementary concern that DualMap[0] addresses through its dual-objective mapping strategy. These works collectively illustrate the central trade-off: aggressive cache reuse can yield substantial speedups, but naive affinity routing risks load imbalance and queueing delays.

Related Works in Same Category

The following **3 sibling papers** share the same taxonomy leaf node with the original paper:

1. Preble: Efficient Distributed Prompt Scheduling for LLM Serving

Authors: Srivatsa, Vikranth, He, Zijian, Vikranth Srivatsa, et al. (14 authors total) | **Year/Venue**: 2024 • International Conference on Learning Representations | **URL**: View paper

Abstract

Prompts to large language models (LLMs) have evolved beyond simple user questions. For LLMs to solve complex problems, today's practices are to include domain-specific instructions, illustration of tool usages, and/or long context such as textbook chapters in

prompts. As such, many parts of prompts are repetitive across requests. Recent works propose to cache and reuse KV state of prompts. However, they are all confined to a single-GPU optimization, while production LLM serving systems are distr...

Relationship Analysis

Both papers belong to the Cache-Affinity and Load-Balancing Routing category, addressing the fundamental trade-off between co-locating requests with shared prefixes for KV cache reuse and distributing load evenly across instances in distributed LLM serving. While DualMap proposes a dual-mapping strategy using two hash functions with SLO-aware routing and hotspot-aware rebalancing to simultaneously achieve both objectives, Preble focuses on a hierarchical scheduling mechanism with a distributed scheduling algorithm that co-optimizes KV state reuse and load-balancing through prompt-aware routing decisions. The key difference is that DualMap employs a power-of-two-choices approach with dual hash rings for elastic scaling, whereas Preble emphasizes hierarchical scheduling across distributed instances with prefix-based caching optimization.

---

2. Online Context Caching for Distributed Large Language Models Serving

Authors: Bin Gao, Zhuomin He, Yizhen Yao, Zhi Zhou, Zhanzhi Lou Lou, et al. (7 authors total) | **Year/Venue**: 2025 | **URL**: View paper

Abstract

â¦ in distributed LLM serving, we model it as an online optimization problem that simultaneously determines KV cache placement and inference request scheduling. â¦ KV Cache Reuse of â¦

Relationship Analysis

Both papers belong to the Cache-Affinity and Load-Balancing Routing category, addressing the fundamental trade-off between co-locating requests with shared prefixes for KV cache reuse and distributing load evenly across instances in distributed LLM serving. While DualMap proposes a dual-mapping strategy using two hash functions to simultaneously achieve cache affinity and load balancing with SLO-aware routing and hotspot-aware rebalancing, the candidate paper frames the problem as an online optimization problem that jointly determines KV cache placement and request scheduling. The key difference is that DualMap uses a dual-hash routing approach with dynamic switching between cache-affinity and load-aware strategies, whereas the candidate paper appears to formulate a unified optimization framework for cache placement and scheduling decisions.

---

3. BanaServe: Unified KV Cache and Dynamic Module Migration for Balancing Disaggregated LLM Serving in AI Infrastructure

Authors: He, Yiyuan, Xu, Minxian, Yiyuan He, et al. (23 authors total) | **Year/Venue**: 2025 | **URL**: View paper

Abstract

Large language models (LLMs) are increasingly deployed in AI infrastructure, driving the need for high throughput, resource efficient serving systems. Disaggregated LLM serving, which separates prompt prefill from auto-regressive decode, has emerged as a promising architecture by isolating their heterogeneous compute and memory demands. However, current disaggregated systems face three key limitations: (i) static resource allocation cannot adapt to highly dynamic workloads, causing over-provisio...

Relationship Analysis

Both papers belong to the Cache-Affinity and Load-Balancing Routing category, addressing the fundamental trade-off between co-locating requests with shared prefixes for cache reuse and distributing load evenly across instances in distributed LLM serving. While DualMap proposes a dual-mapping strategy using two hash functions to simultaneously enable cache affinity and load balancing with SLO-aware routing and hotspot-aware rebalancing, BanaServe focuses on disaggregated architectures (separating prefill and decode stages) and introduces dynamic weight/KV cache migration mechanisms with a Global KV Cache Store to decouple resource allocation from state management. The key distinction is that DualMap operates within a unified serving architecture using dual hash-based routing, whereas BanaServe addresses load imbalance in prefill-decode disaggregated systems through fine-grained resource migration and global cache sharing.

Contributions Analysis

Overall novelty summary. The paper proposes DualMap, a dual-mapping scheduling strategy that simultaneously pursues cache affinity and load balancing in distributed LLM serving. It resides in the 'Cache-Affinity and Load-Balancing Routing' leaf, which contains four papers total (including this one). This leaf sits within the broader 'Request Scheduling and Routing Strategies' branch, indicating a moderately populated research direction. The taxonomy shows that cache-affinity routing is an active area, with sibling works like Preble and BanaServe addressing similar trade-offs between prefix reuse and server utilization.

The taxonomy reveals several neighboring research directions. Adjacent leaves include 'Preemptive and Priority-Based Scheduling' (focusing on fine-grained preemption) and 'SLO-Aware and Adaptive Scheduling' (emphasizing latency guarantees under varying workloads). The broader 'Request Scheduling and Routing Strategies' branch excludes KV cache storage mechanisms, which are handled under 'KV Cache Management and Storage Architectures'. DualMap's dual-mapping approach connects to load-balancing concerns in 'Multi-GPU and Distributed Memory Management' but remains distinct by focusing on request routing rather than memory allocation or hardware heterogeneity.

Among the 30 candidates examined, none clearly refute any of DualMap's three contributions: the dual-mapping strategy itself, the SLO-aware routing technique, and the hotspot-aware rebalancing strategy. Each contribution was assessed against 10 candidates, with zero refutable overlaps identified. This suggests that within the limited search scope, the specific combination of dual hash-based mapping with SLO-aware selection and rebalancing appears novel. However, the analysis does not claim exhaustive coverage; sibling papers like Preble and BanaServe address overlapping concerns (cache affinity, load balancing) using different mechanisms, indicating that the problem space is well-explored even if the exact solution differs.

Based on the top-30 semantic matches and taxonomy structure, DualMap appears to offer a fresh approach to a recognized challenge in distributed LLM serving. The dual-mapping mechanism distinguishes it from single-mapping strategies in sibling works, though the broader goal of reconciling cache reuse with load distribution is shared across the leaf. The limited search scope means that additional related work may exist beyond the examined candidates, particularly in adjacent scheduling or distributed systems literature not captured by the semantic search.

This paper presents **3 main contributions**, each analyzed against relevant prior work:

Contribution 1: DualMap dual-mapping scheduling strategy

Description: A scheduling approach that maps each request to two candidate instances using two independent hash functions based on the request prompt, then intelligently selects between them. This design increases the likelihood that requests with shared prefixes are co-located while evenly dispersing distinct prefixes across the cluster.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. MemServe: Context Caching for Disaggregated LLM Serving with Elastic Memory Pool

URL: View paper

Brief Assessment

MemServe[2] focuses on memory pool management and context caching for disaggregated LLM serving, not on dual-mapping scheduling strategies for cache affinity and load balancing as proposed in the original paper.

---

2. Apt-Serve: Adaptive Request Scheduling on Hybrid Cache for Scalable LLM Inference Serving

URL: View paper

Brief Assessment

Apt-Serve[9] focuses on hybrid cache schemes and adaptive batch scheduling for LLM inference, not dual-mapping strategies for distributed request routing with cache affinity.

---

3. Seesaw: High-throughput llm inference via model re-sharding

URL: View paper

Brief Assessment

Seesaw[58] focuses on dynamic model re-sharding for LLM inference parallelization strategies (tensor vs. pipeline parallelism), not on dual-mapping scheduling for cache affinity and load balancing in distributed serving.

---

4. Mooncake: A kvcache-centric disaggregated architecture for llm serving

URL: View paper

Brief Assessment

Mooncake[12] focuses on a disaggregated KVCache architecture with global cache management and scheduler design for throughput optimization under SLOs. It does not describe a dual-mapping strategy using two independent hash functions for candidate instance selection as proposed in the original paper.

---

5. SkyWalker: A Locality-Aware Cross-Region Load Balancer for LLM Inference

URL: View paper

Brief Assessment

SkyWalker[60] focuses on cross-region load balancing for multi-region LLM deployments with diurnal traffic patterns, not on dual-mapping scheduling within a single cluster. The candidate addresses a different problem scope (multi-region vs. single-cluster distributed serving).

---

6. ServerlessLLM: Low-Latency Serverless Inference for Large Language Models

URL: View paper

Brief Assessment

ServerlessLLM[56] focuses on checkpoint loading and live migration for serverless LLM inference, not on dual-mapping scheduling strategies for distributed serving with cache affinity and load balancing.

---

7. Large Language Model partitioning for low-latency inference at the edge

URL: View paper

Brief Assessment

Edge Partitioning[55] addresses transformer model partitioning across edge devices for inference, focusing on attention head-level allocation to manage memory and compute constraints. This is fundamentally different from DualMap's dual-mapping scheduling strategy for distributed LLM serving, which focuses on request routing with cache affinity and load balancing using two hash functions to map requests to candidate instances.

---

8. PLAIN: Leveraging High Internal Bandwidth in PIM for Accelerating Large Language Model Inference via Mixed-Precision Quantization

URL: View paper

Brief Assessment

PLAIN[57] focuses on hardware acceleration for LLM inference using processing-in-memory (PIM) with mixed-precision quantization. It does not address distributed LLM serving, request scheduling, cache affinity, or load balancing strategies.

---

9. The effect of scheduling and preemption on the efficiency of llm inference serving

URL: View paper

Brief Assessment

Scheduling and Preemption[59] focuses on preemption policies and batch scheduling for LLM inference serving, not on dual-mapping strategies for distributed serving with cache affinity and load balancing across multiple instances.

---

10. Locality-aware Fair Scheduling in LLM Serving

URL: View paper

Brief Assessment

Locality-aware Fair Scheduling[61] focuses on fair scheduling among multiple clients with prefix locality considerations, not on the dual-mapping strategy that maps requests to two candidate instances using independent hash functions for distributed LLM serving.

Contribution 2: SLO-aware request routing technique

Description: A routing strategy that prioritizes prompt-aware scheduling to achieve cache affinity and minimize recomputation overhead, but dynamically shifts to load-aware scheduling only when expected TTFT exceeds the predefined SLO, enhancing load balance without sacrificing cache reuse.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Blockllm: Multi-tenant finer-grained serving for large language models

URL: View paper

Brief Assessment

Blockllm[50] focuses on multi-tenant LLM serving through block-level model partitioning and component sharing, not on SLO-aware routing that balances cache affinity with load-aware scheduling for distributed LLM serving systems.

---

2. Apt-Serve: Adaptive Request Scheduling on Hybrid Cache for Scalable LLM Inference Serving

URL: View paper

Brief Assessment

Apt-Serve[9] addresses batch composition optimization under SLO constraints, not request routing between candidate instances balancing cache affinity and load-aware scheduling.

---

3. PATCHEDSERVE: A Patch Management Framework for SLO-Optimized Hybrid Resolution Diffusion Serving

URL: View paper

Brief Assessment

PATCHEDSERVE[52] focuses on patch-level scheduling for diffusion models in image generation, not LLM serving with KV cache management. The technical domains and optimization targets differ fundamentally.

---

4. Designing Retrieval-Augmented Generation (RAG) Pipelines in Microservice Architectures

URL: View paper

Brief Assessment

RAG Pipelines[53] focuses on SLO-driven orchestration for RAG systems (retrieval, reranking, caching, model selection), not on LLM serving request routing that balances cache affinity with load-aware scheduling for distributed inference clusters.

---

5. AIBrix: Towards Scalable, Cost-Effective Large Language Model Inference Infrastructure

URL: View paper

Brief Assessment

AIBrix[49] focuses on prefix-cache-aware routing and least-latency strategies for LLM gateway optimization, not on the specific dual-mapping approach with SLO-aware switching between cache-affinity and load-aware scheduling described in the original paper.

---

6. A Scalable Approach to Distributed Large Language Model Inference

URL: View paper

Brief Assessment

Scalable Distributed Inference[7] focuses on decoupling KV caches from serving engines and does not present an SLO-aware routing strategy that balances cache affinity with load-aware scheduling based on TTFT thresholds.

---

7. LLM Inference Scheduling: A Survey of Techniques, Frameworks, and Trade-offs

URL: View paper

Brief Assessment

Inference Scheduling Survey[31] is a survey paper that discusses scheduling techniques broadly. The provided context fragments mention 'request scheduling' and 'kv cache reuse' but do not contain sufficient detail about specific SLO-aware routing mechanisms that balance cache affinity and load-aware scheduling to refute the novelty of the original paper's contribution.

---

8. AccelGen: Heterogeneous SLO-Guaranteed High-Throughput LLM Inference Serving for Diverse Applications

URL: View paper

Brief Assessment

AccelGen[54] focuses on iteration-level SLO guarantees for mixed-prompt scenarios with heterogeneous applications, while the original paper addresses cache affinity versus load balancing trade-offs in distributed LLM serving with KV cache reuse.

---

9. SkyLB: A Locality-Aware Cross-Region Load Balancer for LLM Inference

URL: View paper

Brief Assessment

SkyLB[46] focuses on cross-region load balancing with prefix-aware routing and selective pushing mechanisms, not on SLO-aware routing that dynamically switches between cache-affinity and load-aware scheduling based on TTFT thresholds.

---

10. Serving Heterogeneous LoRA Adapters in Distributed LLM Inference Systems

URL: View paper

Brief Assessment

Heterogeneous LoRA Serving[51] addresses SLO-aware routing for LoRA adapters in distributed LLM systems, focusing on rank heterogeneity and adapter placement. The original paper's contribution targets cache affinity vs. load balancing in general distributed LLM serving with KV cache reuse, which is a different technical context.

Contribution 3: Hotspot-aware rebalancing strategy

Description: A rebalancing mechanism that selectively migrates requests from overloaded instances to their backup instances (the alternative instance from the initial dual mapping), mitigating hotspots and rebalancing the system under skewed prefix popularity workloads.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Arrow: Adaptive Scheduling Mechanisms for Disaggregated LLM Inference Architecture

URL: View paper

Brief Assessment

Arrow[44] focuses on adaptive scheduling in prefill-decode disaggregated architectures for LLM inference, not on distributed serving with dual-mapping and KV cache affinity. The rebalancing mechanisms address different architectural contexts and workload characteristics.

---

2. Ascendra: Dynamic Request Prioritization for Efficient LLM Serving

URL: View paper

Brief Assessment

Ascendra[48] focuses on request prioritization and offloading between low-priority and high-priority GPU instances to meet TTFT and TBT SLOs in LLM serving. The original paper's hotspot-aware rebalancing addresses load imbalance caused by skewed prefix popularity in distributed KV cache systems through selective request migration between dual-mapped backup instances. These are fundamentally different mechanisms operating in different architectural contexts.

3. Temporal-Aware GPU Resource Allocation for Distributed LLM Inference via Reinforcement Learning

URL: View paper

Brief Assessment

Temporal GPU Allocation[42] addresses hotspot mitigation in distributed LLM inference through temporal-aware GPU resource allocation and reinforcement learning-based scheduling. However, its focus is on GPU resource allocation across geographical regions with temporal dependencies, not on KV cache-aware request migration between backup instances in distributed serving systems.

4. WindServe: Efficient Phase-Disaggregated LLM Serving with Stream-based Dynamic Scheduling

URL: View paper

Brief Assessment

WindServe[4] focuses on phase-disaggregated LLM serving with dynamic scheduling between prefill and decoding phases, not on hotspot-aware rebalancing through request migration in distributed systems with dual-mapping.

5. GRACE-MoE: Grouping and Replication with Locality-Aware Routing for Efficient Distributed MoE Inference

URL: View paper

Brief Assessment

GRACE-MoE[47] addresses load imbalance in distributed MoE inference through expert grouping and replication, not request migration between backup instances in LLM serving systems with KV cache considerations.

6. Llumnix: Dynamic scheduling for large language model serving

URL: View paper

Brief Assessment

Llumnix[41] focuses on live migration of requests across LLM instances to address load imbalance, but does not specifically describe a hotspot-aware rebalancing mechanism that selectively migrates requests to backup instances based on dual mapping as described in the original paper.

7. Online Context Caching for Distributed Large Language Models Serving

URL: View paper

Brief Assessment

Online Context Caching[26] focuses on kv cache placement and request scheduling as an online optimization problem in distributed LLM serving. The candidate does not describe a hotspot-aware rebalancing mechanism that selectively migrates requests from overloaded instances to backup instances based on dual mapping, which is the core novelty of the original paper's contribution.

8. {dLoRA}: Dynamically orchestrating requests and adapters for {LoRA}{LLM} serving

URL: View paper

Brief Assessment

dLoRA[45] addresses hotspot-aware rebalancing in the context of LoRA adapter serving with dynamic request migration between replicas. The original paper focuses on distributed LLM serving with KV cache management and prefix-based dual mapping, which is a fundamentally different architectural approach and problem domain.

9. SkyLB: A Locality-Aware Cross-Region Load Balancer for LLM Inference

URL: View paper

Brief Assessment

SkyLB[46] addresses load imbalance through selective pushing based on pending requests across regions, not through migrating requests to backup instances within a dual-mapping framework to handle skewed prefix popularity.

10. DynaServe: Unified and Elastic Execution for Dynamic Disaggregated LLM Serving

URL: View paper

Brief Assessment

DynaServe[43] focuses on dynamic request partitioning and batch composition for LLM serving, not on hotspot-aware rebalancing through selective request migration in distributed systems with dual mapping and backup instances.

Appendix: Text Similarity Detection

Textual similarity detection checked 30 papers and found 2 similarity segment(s) across 1 paper(s).

The following **1 paper(s)** were detected to have high textual similarity with the original paper. These may represent different versions of the same work, duplicate submissions, or papers with substantial textual overlap. Readers are advised to verify these relationships independently.

1. Preble: Efficient Distributed Prompt Scheduling for LLM Serving

Detected in: Core Task (sibling)

⚠ **Note**: This paper shows substantial textual similarity with the original paper. It may be a different version, a duplicate submission, or contain significant overlapping content. Please review carefully to determine the nature of the relationship.

References

- [0] DualMap: Enabling Both Cache Affinity and Load Balancing for Distributed LLM Serving View paper
- [1] Fast State Restoration in LLM Serving with HCache View paper
- [2] MemServe: Context Caching for Disaggregated LLM Serving with Elastic Memory Pool View paper
- [3] Optimizing SLO-oriented LLM Serving with PD-Multiplexing View paper
- [4] WindServe: Efficient Phase-Disaggregated LLM Serving with Stream-based Dynamic Scheduling View paper
- [5] Continuum: Efficient and Robust Multi-Turn LLM Agent Scheduling with KV Cache Time-to-Live View paper
- [6] Layerkv: Optimizing large language model serving with layer-wise kv cache management View paper
- [7] A Scalable Approach to Distributed Large Language Model Inference View paper
- [8] FastCache: Optimizing Multimodal LLM Serving through Lightweight KV-Cache Compression Framework View paper
- [9] Apt-Serve: Adaptive Request Scheduling on Hybrid Cache for Scalable LLM Inference Serving View paper
- [10] FlashInfer: Efficient and Customizable Attention Engine for LLM Inference Serving View paper

- [11] Serving Large Language Models on Huawei CloudMatrix384 View paper
- [12] Mooncake: A kvcache-centric disaggregated architecture for llm serving View paper
- [13] FastDecode: High-Throughput GPU-Efficient LLM Serving using Heterogeneous Pipelines View paper
- [14] Accelerating LLM Serving for Multi-turn Dialogues with Efficient Resource Management View paper
- [15] A System for Microserving of LLMs View paper
- [16] COMET: Towards Practical W4A4KV4 LLMs Serving View paper
- [17] Preble: Efficient Distributed Prompt Scheduling for LLM Serving View paper
- [18] A survey on large language model acceleration based on kv cache management View paper
- [19] Online Scheduling for LLM Inference with KV Cache Constraints View paper
- [20] TokenFlow: Responsive LLM Text Streaming Serving under Request Burst via Preemptive Scheduling View paper
- [21] No Request Left Behind: Tackling Heterogeneity in Long-Context LLM Inference with Medha View paper
- [22] FineServe: Precision-Aware KV Slab and Two-Level Scheduling for Heterogeneous Precision LLM Serving View paper
- [23] CE-LSLM: Efficient Large-Small Language Model Inference and Communication via Cloud-Edge Collaboration View paper
- [24] MCaM : Efficient LLM Inference with Multi-tier KV Cache Management View paper
- [25] Compute or load kv cache? why not both? View paper
- [26] Online Context Caching for Distributed Large Language Models Serving View paper
- [27] DroidSpeak: KV Cache Sharing for Cross-LLM Communication and Multi-LLM Serving View paper
- [28] BanaServe: Unified KV Cache and Dynamic Module Migration for Balancing Disaggregated LLM Serving in AI Infrastructure View paper
- [29] Mell: Memory-Efficient Large Language Model Serving via Multi-GPU KV Cache Management View paper
- [30] KVShare: An LLM Service System with Efficient and Effective Multi-Tenant KV Cache Reuse View paper
- [31] LLM Inference Scheduling: A Survey of Techniques, Frameworks, and Trade-offs View paper
- [32] FlowKV: A Disaggregated Inference Framework with Low-Latency KV Cache Transfer and Load-Aware Scheduling View paper
- [33] Tokencake: A KV-Cache-centric Serving Framework for LLM-based Multi-Agent Applications View paper
- [34] ServerlessPD: Fast RDMA-Codesigned Disaggregated Prefill-Decoding for Serverless Inference of Large Language Models View paper
- [35] Self-Defining Systems View paper
- [36] TARDIS: A GPU-Centric KV Cache Service for Efficient LLM Inference View paper
- [37] xLLM Technical Report View paper
- [38] SPADA: Secure, Performant, and Distributed LLM Inference View paper
- [39] Workshop on Mobility in the Evolving Internet Architecture to be held in conjunction with MobiComï¿ºï¿¹ï¿ºï¿¼ View paper
- [40] Sim-LLM: Optimizing LLM Inference at the Edge through Inter-Task KV Reuse View paper
- [41] Llumnix: Dynamic scheduling for large language model serving View paper
- [42] Temporal-Aware GPU Resource Allocation for Distributed LLM Inference via Reinforcement Learning View paper
- [43] DynaServe: Unified and Elastic Execution for Dynamic Disaggregated LLM Serving View paper
- [44] Arrow: Adaptive Scheduling Mechanisms for Disaggregated LLM Inference Architecture View paper
- [45] {dLoRA}: Dynamically orchestrating requests and adapters for {LoRA}{LLM} serving View paper
- [46] SkyLB: A Locality-Aware Cross-Region Load Balancer for LLM Inference View paper
- [47] GRACE-MoE: Grouping and Replication with Locality-Aware Routing for Efficient Distributed MoE Inference View paper
- [48] Ascendra: Dynamic Request Prioritization for Efficient LLM Serving View paper
- [49] AIBrix: Towards Scalable, Cost-Effective Large Language Model Inference Infrastructure View paper
- [50] Blockllm: Multi-tenant finer-grained serving for large language models View paper
- [51] Serving Heterogeneous LoRA Adapters in Distributed LLM Inference Systems View paper
- [52] PATCHEDSERVE: A Patch Management Framework for SLO-Optimized Hybrid Resolution Diffusion Serving View paper
- [53] Designing Retrieval-Augmented Generation (RAG) Pipelines in Microservice Architectures View paper
- [54] AccelGen: Heterogeneous SLO-Guaranteed High-Throughput LLM Inference Serving for Diverse Applications View paper
- [55] Large Language Model partitioning for low-latency inference at the edge View paper
- [56] ServerlessLLM: Low-Latency Serverless Inference for Large Language Models View paper
- [57] PLAIN: Leveraging High Internal Bandwidth in PIM for Accelerating Large Language Model Inference via Mixed-Precision Quantization View paper
- [58] Seesaw: High-throughput llm inference via model re-sharding View paper
- [59] The effect of scheduling and preemption on the efficiency of llm inference serving View paper
- [60] SkyWalker: A Locality-Aware Cross-Region Load Balancer for LLM Inference View paper
- [61] Locality-aware Fair Scheduling in LLM Serving View paper