# PR #24880 完整报告

sgl-project/sglang

[PD & HiSparse] Add DeepSeek V4 support for HiSparse direct Prefill-to-Decode DRAM

合并时间：2026-06-05 15:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24880

## 执行摘要

- 一句话：支持 DeepSeek V4 HiSparse 直接 PD 主机传输，TTFT 降 7-9%
- 推荐动作：此 PR 涉及推理引擎内核、内存池、JIT 编译和远程传输多个模块的联动修改，建议所有参与 SGLang 推理引擎开发的同学精读。尤其是 DeepSeekV4PagedHostPool 的布局设计、transfer_cache_dsv4_mla 的 JIT 实现、以及 Mooncake 传输的 PP 对齐策略，具有较高的参考价值。

## 功能与动机

在 PD + HiSparse 场景下支持 DeepSeek V4 模型，以提高 C4 KV 缓存的传输效率。之前的实现仅支持非 DSV4 模型，DSV4 的 C4 层具有独特的 MLA 结构，需要专用的传输逻辑。此外，旧的 token 线性主机池布局与 HiSparse 的页语义不一致，导致索引偏移 bug。PR 的性能数据表明，C4 host-transfer 优化使 mean TTFT 降低 7.16%。

## 实现拆解

实现拆解如下：

1. 重构主机内存池布局：在 python/sglang/srt/mem_cache/memory_pool_host.py 中将 DeepSeekV4PagedHostPool 基类从 HostKVCache 改为 HiSparseHostPoolMixin, HostKVCache，新增 get_contiguous_buf_infos() 方法返回每层页行缓冲区用于直接传输，新增 _has_transfer_indices() 方法统一检查索引有效性。在 backup_from_device_all_layer() 中添加非页对齐（token 粒度）的分支，调用新 JIT 函数处理 DSV4 MLA 内容。
2. 迁移和集中 JIT 内核：删除 sglang/jit_kernel/dsv4/hisparse.py 和 csrc/deepseek_v4/hisparse_transfer.cuh（旧 JIT 封装和 CUDA 内核）。在通用 sglang/jit_kernel/hisparse.py 中新增 _jit_dsv4_transfer_module 和 transfer_cache_dsv4_mla 函数，实现页内 token 拷贝（处理 value/scale 非连续布局）。
3. 更新 Mooncake 传输通道：在 mooncake/conn.py 中添加 _send_kvcache_hisparse_dsv4 和 _send_dsv4_state 方法。前者将 C4 KV 通过新 JIT 函数写入主机，同时将 c4_indexer/c128 通过原有 _send_kvcache_generic 传输；后者复用 SWA 状态发送逻辑。PP 对齐通过 get_mla_kv_ptrs_with_pp 统一处理。
4. 启用直接 PD admission：在 python/sglang/srt/managers/hisparse_coordinator.py 中移除 admit_request_direct 的 NotImplementedError，改用

DeepSeekV4PagedHostPool 实例替换 DeepSeekV4SingleKVPoolHost，并新增 host_token_len() 方法适配压缩比，使 preload 路径正确工作。

5. 补充测试：在 python/sglang/jit_kernel/tests/test_hisparse.py 中添加 test_transfer_cache_dsv4_mla_copies_paged_token（验证页内 token 拷贝）和 test_dsv4_swap_in_reads_paged_host_layout（验证 swap-in 读取）；在 test/registered/disaggregation/test_disaggregation_dsv4.py 中添加 TestDisaggregationDSV4HiSparseMooncake 端到端测试，GSM8K 200 题评估。

关键文件：

- python/sglang/srt/mem_cache/memory_pool_host.py（模块 内存池；类别 source；类型 core-logic；符号 DeepSeekV4PagedHostPool, get_contiguous_buf_infos, _has_transfer_indices）：核心逻辑：DeepSeekV4PagedHostPool 基类变更，新增 get_contiguous_buf_infos 和 _has_transfer_indices，backup_from_device_all_layer 添加 DSV4 专用分支。
- python/sglang/srt/mem_cache/hisparse_memory_pool.py（模块 分配器；类别 source；类型 dependency-wiring；符号 DeepSeekV4SingleKVPoolHost）：移除旧类 DeepSeekV4SingleKVPoolHost 及 psutil 依赖，简化 host 池初始化。
- python/sglang/jit_kernel/hisparse.py（模块 JIT 内核；类别 source；类型 core-logic；符号 _jit_dsv4_transfer_module, transfer_cache_dsv4_mla）：新增 JIT 函数 transfer_cache_dsv4_mla，处理 DSV4 MLA 页内 token 拷贝。
- python/sglang/srt/managers/hisparse_coordinator.py（模块 协调器；类别 source；类型 dependency-wiring；符号 host_token_len）：启用直接 PD admission，替换主机池实现，新增 host_token_len 方法。
- test/registered/disaggregation/test_disaggregation_dsv4.py（模块 集成测试；类别 test；类型 test-coverage；符号 TestDisaggregationDSV4HiSparseMooncake, setUpClass, start_prefill, start_decode）：新增端到端集成测试 TestDisaggregationDSV4HiSparseMooncake，确保 GSM8K 准确率达 0.93。

关键符号：DeepSeekV4PagedHostPool.get_contiguous_buf_infos, DeepSeekV4PagedHostPool._has_transfer_indices, DeepSeekV4PagedHostPool.backup_from_device_all_layer, transfer_cache_dsv4_mla, HiSparseCoordinator.host_token_len, HiSparseCoordinator.admit_request_direct, MooncakeConn._send_kvcache_hisparse_dsv4, MooncakeConn._send_dsv4_state, test_transfer_cache_dsv4_mla_copies_paged_token, test_dsv4_swap_in_reads_paged_host_layout

## 关键源码片段

### python/sglang/srt/mem_cache/memory_pool_host.py

核心逻辑：DeepSeekV4PagedHostPool 基类变更，新增 get_contiguous_buf_infos 和 _has_transfer_indices，backup_from_device_all_layer 添加 DSV4 专用分支。

```
# memory_pool_host.py — DeepSeekV4PagedHostPool 关键新增方法
class DeepSeekV4PagedHostPool(HiSparseHostPoolMixin, HostKVCache):
```

```
def get_contiguous_buf_infos(self):
    """返回每层页行缓冲区信息，用于 PD 直连传输"""
    data_ptrs = [int(self.data_ptrs[i].item()) for i in range(self.layer_num)]
    data_lens = [self.kv_buffer[i].nbytes for i in range(self.layer_num)]
    item_lens = [self.item_bytes * self.dtype.itemsize] * self.layer_num
    return data_ptrs, data_lens, item_lens

def _has_transfer_indices(
    self, host_indices: torch.Tensor | None, device_indices: torch.Tensor | None
) -> bool:
    # 统一校验两个索引是否存在且长度匹配
    if host_indices is None or device_indices is None:
        return False
    if host_indices.numel() != device_indices.numel():
        raise ValueError("索引大小不匹配")
    return host_indices.numel() > 0

def backup_from_device_all_layer(
    self, device_pool, host_indices, device_indices, io_backend
):
    if not self._has_transfer_indices(host_indices, device_indices):
        return
    if ( # 非页对齐时使用 token 粒度的 DSV4 拷贝
        host_indices.numel() % self.slot_page_size != 0
        or device_indices.numel() % self.slot_page_size != 0
    ):
        # DSV4 MLA 的 token 在页内不是连续字节，需专用 JIT 函数
        transfer_cache_dsv4_mla(
            src_ptrs=self.device_ptrs,
            dst_ptrs=self.data_ptrs,
            src_indices=device_indices.to(torch.int64),
            dst_indices=host_indices.to(torch.int64),
        )
        return
    # 页对齐时走原有快速路径
    host_rows = self._to_page_indices(host_indices)
    device_rows = self._to_page_indices(device_indices)
    transfer_kv_all_layer_mla(
        src_layers=self.device_ptrs,
        dst_layers=self.data_ptrs,
        src_indices=device_rows,
        dst_indices=host_rows,
        ...
    )
```

# 评论区精华

关键讨论摘要：

- 性能优化：gemini-code-assist 建议聚合跨层传输块以减少同步调用次数，避免多次 _send_kvcache_generic 调用；使用 clip 而非断言处理状态索引长度不匹配；用 CUDA events 替代 synchronize() 以避免阻塞调度循环。这些建议被部分采纳（如 PP 对齐处理）。
- PP 对齐正确性：ShangmingCai 关注 PP+PD 下设备 KV 指针切片的正确性，作者最终通过 get_mla_kv_ptrs_with_pp 对齐处理并拆分 C4 和其他部分，确保 PP 分片正确。
- Sentinel slot 变更：ShangmingCai 对 DeepSeekV4SingleKVPoolHost 中 free_slots 起始值从 1 改为 0 提出疑问。作者解释 host 池与 device 池 namespace 不同，slot 0 在 host 池中有效，旧实现因页索引展开导致偏移 1 的 bug。
- 测试覆盖：hzh0425 要求添加 PD + DSV4 HiSparse 的端到端测试，作者回应该请求并添加了 GSM8K 集成测试。
  - Mooncake 传输性能优化 (performance): 部分采纳，通过 PP 对齐统一处理；后续可进一步优化。
  - 状态索引断言安全性 (correctness): 未采纳，因为索引由内部逻辑保证一致，但 reviewer 认可防御性编程的合理性。
  - 同步方式选择 (performance): 未修改，因为当前 synchronize 在 admission 冷路径中，影响有限。
  - PP 对齐正确性 (design): 通过使用 get_mla_kv_ptrs_with_pp 对齐后拆分 C4 和其他部分，确保正确。
  - Sentinel slot 初始值 (question): 作者解释 host 池与 device 池 namespace 不同，slot 0 在 host 池中有效，旧偏移导致 bug。

## 风险与影响

- 风险：风险分析：
  - 兼容性风险：删除了 DeepSeekV4SingleKVPoolHost 类和旧 JIT 内核 hisparse_offload_to_host，任何外部代码直接引用这些符号将 break。但该模块主要用于内部管理调度，且已由功能等价的 DeepSeekV4PagedHostPool 和 transfer_cache_dsv4_mla 替代。
  - 性能退化：新的分页布局可能引入额外页计算开销，但基准测试显示 TTFT 改善（均值和中位数均下降），未发现退化。
  - 正确性风险：PP+PD 场景下设备指针切片逻辑复杂，首次实现时可能存在遗漏，经 review 迭代后已确认使用 get_mla_kv_ptrs_with_pp 统一对齐。
  - 测试覆盖风险：端到端测试仅基于 DeepSeek V4 Flash 模型（FP8），未覆盖 Pro 或其他变体，但 Flash 已是主力推理模型。
- 影响：影响分析：
  - 用户：使用 DeepSeek V4 Flash/Pro 并开启 PD + HiSparse 的用户将获得显著的 TTFT 降低（7-9%），且无需额外配置。
  - 系统：主机内存消耗可能因页对齐预留而略有增加，但通过压缩比控制；Mooncake 传输逻辑复杂度上升，增加维护成本。
  - 团队：后续需将主机池从 token 语义升级为页语义（已记录 follow-up TODO），以进一步优化内存管理。

- 风险标记：核心路径变更，兼容性风险，测试覆盖新增，性能优化收益

## 关联脉络

- 暂无明显关联 PR