PR #25477 完整报告

sgl-project/sglang

[BugFix]: Fix DeepSeek V4 HiCache layer count logic

合并时间：2026-05-16 23:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25477

执行摘要

- 一句话：修复 DeepSeek V4 HiCache 层计数逻辑并拆分测试 CI
- 推荐动作：此 PR 修复了关键的 PP + HiCache 兼容性问题，核心逻辑改动集中在层映射计算，值得精读以理解 PP 对缓存层的影响。同时应关注两个遗留风险：state pools 索引偏移和测试空覆盖，建议在后续 PR 中跟进修复。

功能与动机

PP 支持提交改变了 layer_mapping 结构，使其包含 None 条目，原有的 len(compression_ratios) 不再代表真实的活跃层数，导致 HiCache 在多层映射、sidecar pool 注册时访问到不存在的层，引发崩溃或数据错误。PR body 明确说明：“A previous PP support commit changed DeepSeekV4TokenToKVPool.layer_mapping to be stage-aware, leaving entries outside the active layer range as None.”

实现拆解

1. 核心逻辑修复（hybrid_pool_assembler.py）：将 build_deepseek_v4_hicache_stack 和 attach_hybrid_pool_to_unified_cache 中的 transfer_layer_num 从 len(kvcache.compression_ratios) 改为 kvcache.end_layer - kvcache.start_layer，并添加 TODO 备注后续支持 PP。
2. 层映射切片修正：在枚举 kvcache.layer_mapping 构建 C4/C128 映射时，增加切片 [kvcache.start_layer : kvcache.end_layer]，确保只处理当前 PP stage 拥有的层，避免访问 None 条目。
3. 测试拆分与 CI 重排：
   - 新增 test_unified_radix_cache_kl_hicache.py，将原本仅在 nightly 中运行的 Mamba 和 DeepSeek V4 HiCache 测试移至 base CI，注册为 base-c stage。
   - 原 test_unified_radix_hicache_kl.py 重命名为 test_unified_radix_cache_kl_hicache_nightly.py，移除 Mamba 和 DeepSeek V4 测试，仅保留 GLM5 模型测试，并新增 GSM8KTwoPassMixin 用于验证精度稳定性。

关键文件：

- python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py（模块 缓存层；类别 source；类型 core-logic；符号 build_deepseek_v4_hicache_stack, attach_hybrid_pool_to_unified_cache）：核心修复文件：修改了 DeepSeek V4 HiCache

的 transfer_layer_num 计算和层映射枚举逻辑，确保 PP 场景下只处理当前 stage 的活跃层。

- test/registered/radix_cache/test_unified_radix_cache_kl_hicache.py（模块 测试；类别 test；类型 test-coverage；符号 TestUnifiedMambaHiCache, setUpClass, tearDownClass, _assert_dsv4_decode_cached_tokens）：新增的 base CI 测试文件，将 Mamba 和 DeepSeek V4 HiCache 测试从 nightly 迁移过来，注册在 base-c stage，确保日常 CI 覆盖。
- test/registered/radix_cache/test_unified_radix_cache_kl_hicache_nightly.py（模块 测试；类别 test；类型 rename-or-move；符号 TestUnifiedMambaHiCache, setUpClass, tearDownClass, _assert_dsv4_decode_cached_tokens）：重命名并精简后的 nightly 测试文件，仅保留 GLM5 模型测试，并添加 GSM8KTwoPassMixin 以验证精度稳定性。

关键符号：build_deepseek_v4_hicache_stack, attach_hybrid_pool_to_unified_cache

关键源码片段

python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py

核心修复文件：修改了 DeepSeek V4 HiCache 的 transfer_layer_num 计算和层映射枚举逻辑，确保 PP 场景下只处理当前 stage 的活跃层。

```
# python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py

def build_deepseek_v4_hicache_stack(
    *,
    params: CacheInitParams,
    server_args: ServerArgs,
    kvcache: Any,
    page_size: int,
    tp_group,
    load_cache_event,
    attn_cp_group=None,
    attn_tp_group=None,
    storage_backend: Optional[str],
    host_swa_evict_fn=None,
    device_swa_evict_fn=None,
    prefetch_threshold: int = 256,
    model_name: Optional[str] = None,
    storage_backend_extra_config: Optional[dict] = None,
    pp_rank: int = 0,
    pp_size: int = 1,
    enable_storage_metrics: bool = False,
) -> tuple[HostPoolGroup, HybridCacheController]:
    # 使用 end_layer - start_layer 替代 len(compression_ratios)
    # 确保在多 PP stage 下只计算当前 stage 拥有的层数
    transfer_layer_num = kvcache.end_layer - kvcache.start_layer

    full_layer_mapping = {layer_id: layer_id for layer_id in range(transfer_layer_num)}
```

```
swa_layer_mapping = {
    layer_id: layer_id for layer_id in range(len(kvcache.swa_kv_pool.kv_buffer))
}

c4_layer_mapping = {}
c128_layer_mapping = {}
c4_state_global_layers = []
c128_state_global_layers = []
# 对 layer_mapping 切片，只处理当前 stage 的层（避免访问 None 条目）
for layer_id, layer_item in enumerate(
    kvcache.layer_mapping[kvcache.start_layer : kvcache.end_layer]
):
    if layer_item.compress_ratio == 4:
        c4_layer_mapping[layer_id] = layer_item.compress_layer_id
        c4_state_global_layers.append(layer_id)
    elif layer_item.compress_ratio == 128:
        c128_layer_mapping[layer_id] = layer_item.compress_layer_id
        c128_state_global_layers.append(layer_id)
# ... 后续使用 c4_state_global_layers 索引 kvcache.compress_state_pools
# 注意：review 指出此处 layer_id 为局部索引，可能引起 PP 下 state pools 访问错误
```

评论区精华

gemini-code-assist[bot] 在 review 中指出了两个高优问题：

- state pools 索引风险：在 hybrid_pool_assembler.py 中，切片后 layer_id 从 0 开始计数，但 c4_state_global_layers 仍用此局部索引访问全局 compress_state_pools，在 PP 模式下会访问错误的 state pools。该评论标记为高优先级，但未在后续提交中修复。
- 测试空覆盖：新增的测试类中存在重复的 test_gsm8k 定义，且都使用 pass 实现，导致继承自 UnifiedRadixTreeTestMixin 的实际测试被静默禁用，测试套件会误报成功。建议改用 @unittest.skipIf 装饰器调用 super()。

ShangmingCai 最后批准了 PR，评论“LGTM, let's wait for the CI.”

- State pools 索引风险 (correctness): 未在代码中修复，PR 仍被批准合并。
- 测试类中重复定义 test_gsm8k 并使用 pass (testing): 未修复，PR 被批准合并。
- TestUnifiedDeepSeekV4FlashHiCache 同样问题 (testing): 未修复，PR 被批准合并。

风险与影响

- 风险：
  1. 未完全解决的 state pools 索引风险：review 指出的使用局部索引访问全局 state pools 的问题在 PP 模式下可能仍会导致错误，虽然当前 kvcache.compress_state_pools 的结构可能恰好使偏移正确，但需要警惕。
  2. 测试中 pass 覆盖风险：TestUnifiedMambaHiCache 和 TestUnifiedDeepSeekV4FlashHiCache 存在重复的 test_gsm8k 定义（均为 pass），导致 GSM8K 精度测试在新增的 base CI 中不会实际执行，可能隐藏回归。

3. CI 执行时间：将原本 nightly 的 Mamba/DSV4 测试移至 base CI 增加了 base-c stage 的执行时间（估计 768 秒），但通过拆分文件避免全部 nightly 测试一起运行，整体 CI 时间可控。 - 影响：用户影响：修复了 DeepSeek V4 使用 HiCache + PP 时的启动崩溃或推理错误，用户升级后可正常启用 hierarchical cache。系统影响：CI 结构变化，base CI 新增两个大型测试（Mamba 和 DSV4），nightly 测试减少，有助于更快发现回归。团队影响：需维护两个测试文件，且注意修复 review 中遗留的问题。

- 风险标记：state pools 索引风险未修复，测试空覆盖可能隐藏回归

关联脉络

- PR #24704 feat: add Pipeline Parallelism (PP) and PD support for DeepSeek-V4: PP 支持改变了 DeepSeekV4TokenToKVPool.layer_mapping 结构，导致 HiCache 层计数失效，是此 bug 的直接诱因。