PR #25889 完整报告

sgl-project/sglang

[Fix] DSV4 cached_loc invalidated when SWA mapping is rebuilt

合并时间：2026-05-21 13:38

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25889

执行摘要

- 一句话：修复 DSV4 cached_loc 在映射重建后未失效
- 推荐动作：值得精读，尤其是测试分层设计——从协议桩（stub）到崩溃回归再到端到端 KL 验证，展示了高质量的防御性编程。适合作为缓存失效类 bug 的修复范本。

功能与动机

当 SGLANG_OPT_CACHE_SWA_TRANSLATION=True 时，cached_loc 优化可能返回过期的翻译结果，因为 register_mapping 替换 full_to_swa_index_mapping 后没有清除 cached_loc。这会在 HiCache commit/load-back 重建映射后引发 KV 写入错误槽位，产生 logprobs 差异或 IndexError。PR body 中描述了错误日志和复现场景。

实现拆解

1. 核心修复：在 deepseek_v4_memory_pool.py 的 register_mapping 方法中添加 self.cached_loc = None，使每次映射更新时立即清除缓存。
2. 新增公共方法：覆盖父类 BaseSWAKVPool 的 invalidate_loc_cache 空实现，使其真正清空 cached_loc，配合 model_runner._forward_raw 阶段调用。
3. 单元测试：
   - test_dsv4_cached_loc_invalidation.py：使用 _DSV4CacheStub 模拟缓存逻辑，验证修复前返回过期值、修复后返回新值。
   - test_dsv4_stale_loc_crash.py：使用 _SWAPoolMock 模拟实际 SWA 池写入，验证过期 cached_loc 导致 IndexError，修复后正常。
4. 端到端测试：test_dsv4_hicache_swa_translation_cache.py 子类化 UnifiedRadixTreeTestMixin，启动 DSV4 Flash FP8 服务器并启用 SGLANG_OPT_CACHE_SWA_TRANSLATION=1，通过 KL 散度检查跨 HiCache 加载的正确性。 以上测试均位于 test/manual/core/，需手动运行（因耗时或环境要求未纳入 CI 自动套件）。

关键文件：

- python/sglang/srt/mem_cache/deepseek_v4_memory_pool.py（模块 内存池；类别 source；类型 core-logic；符号 invalidate_loc_cache）：核心源文件，修复 register_mapping 未清除 cached_loc 的 bug，并新增 invalidate_loc_cache 方法。

- test/manual/core/test_dsv4_cached_loc_invalidation.py（模块 单元测试；类别 test；类型 test-coverage；符号 _DSV4CacheStub, init, register_mapping_buggy, register_mapping_fixed）：使用 _DSV4CacheStub 复现缓存过期 bug，并验证修复逻辑。
- test/manual/core/test_dsv4_stale_loc_crash.py（模块 回归测试；类别 test；类型 test-coverage；符号 _SWAPoolMock, init, set_key_buffer_fused, _build_pool）：模拟真实池写入，验证过期 cached_loc 导致 IndexError，修复后不再崩溃。
- test/manual/core/test_dsv4_hicache_swa_translation_cache.py（模块 E2E 测试；类别 test；类型 test-coverage；符号 TestDSV4HiCacheSWATranslationCache, test_multiturn_logprobs_match, setUpClass, tearDownClass）：端到端 KL 回归测试，在真实服务器环境中验证 HiCache 与 SWA 缓存交互的正确性。

关键符号：register_mapping, invalidate_loc_cache, get_swa_loc, set_swa_key_buffer_radix_fused

关键源码片段

python/sglang/srt/mem_cache/deepseek_v4_memory_pool.py

核心源文件，修复 register_mapping 未清除 cached_loc 的 bug，并新增 invalidate_loc_cache 方法。

```
# register_mapping: 替换映射时清除缓存
self.cached_loc = None

def register_mapping(self, full_to_swa_index_mapping: torch.Tensor):
    self.full_to_swa_index_mapping = full_to_swa_index_mapping
    # 清除缓存的 SWA 索引，防止后续使用过期的映射
    self.cached_loc = None

def invalidate_loc_cache(self) -> None:
    """每个批次前由 model_runner 调用，清空缓存以确保正确性"""
    self.cached_loc = None
```

评论区精华

此 PR 未触发实质性人工 review 讨论。gemini-code-assist[bot] 自动审查确认了修改逻辑正确，未提出额外反馈。

- 暂无高价值评论线程

风险与影响

- 风险：修复本身仅添加一行 self.cached_loc = None，风险极低。主要风险在于：若未来引入新的 cached_loc 赋值路径却未同步清理，可能再次出现类似 bug。另外，invalidate_loc_cache 的覆盖依赖于 model_runner 的调用约定，若调用顺序或条件发生变化，可能失效。测试覆盖了三层，但均为手动运行，可能因环境差异遗漏。
- 影响：直接影响范围：仅使用 DeepSeek V4 模型且显式设置环境变量 SGLANG_OPT_CACHE_SWA_TRANSLATION=True 的用户。修复后，HiCache 提交 / 加

载后 SWA 索引缓存正确失效，避免 KV 写入错误和崩溃。对其他配置的用户无影响。团队开发需注意类似“缓存未随依赖失效”的模式。

- 风险标记：缓存未随依赖失效，默认关闭影响面小

关联脉络

- PR #25824 [Refactor] Encapsulate SWA loc translation inside SWAKVPool with per-batch cache invalidation: 该 PR 引入了 SWA 位置翻译缓存机制，但遗漏了 register_mapping 时的缓存失效，本 PR 是直接补充修复。
- PR #25646 fix deepseek v4 hisparse: 同为 DeepSeek V4 的 bugfix，涉及压缩器逻辑，与 SWA 缓存无直接关联但属于同一模型功能的稳定性改进。
- PR #24226 [BugFix] fix(hicache): fix two slot-reuse races in DecodeKVCacheOffloadManager: 同为 HiCache 相关 bugfix，体现了 HiCache 功能在生产中逐步完善的趋势。