PR #24779 完整报告

sgl-project/sglang

fix UnifiedRadixCache MTP child_key index out of range

合并时间：2026-05-09 23:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24779

执行摘要

- 一句话：修复 MTP 场景下 Radix Cache 越界崩溃
- 推荐动作：建议快速合并，这是一个明确的边界条件崩溃修复，改动小且逻辑清晰。值得学习的是使用缓存对象避免重复创建空 tensor 的模式，减少内存分配和 GC 压力。

功能与动机

启动 DeepSeek V4 搭配 HiCache 和 MTP 时，调度器因 UnifiedRadixCache.match_prefix 中 key 为空导致 child_key 越界崩溃。PR body 中给出了完整的堆栈跟踪，根因是 match_prefix 在 key 长度为 0 但 page_aligned 后仍可能保持 0 时，未提前返回，而进入了 _match_prefix_helper 导致访问越界。

实现拆解

1. 缓存空匹配结果：在 _reset_full 方法末尾新增 self._empty_match_result，预先创建 MatchResult 实例，其中 device_indices 为空 tensor、last_device_node 和 last_host_node 设为 self.root_node。
2. 提前返回空结果：在 match_prefix 方法中（涉及 python/sglang/srt/mem_cache/unified_radix_cache.py），当 key 长度为 0 时直接返回 self._empty_match_result，不再创建临时 MatchResult。在 key.page_aligned(self.page_size) 后增加 if len(key) == 0: return self._empty_match_result，确保 page_aligned 后空 key 不进入 _match_prefix_helper。
3. 统一空结果引用：在 _match_post_processor 和 init_load_back 中，将原本直接 torch.empty(...) 创建空 tensor 的地方替换为 self._empty_match_result.device_indices，减少重复 tensor 创建。

关键文件：

- python/sglang/srt/mem_cache/unified_radix_cache.py（模块 缓存层；类别 source；类型 core-logic；符号 _reset_full, match_prefix, _match_post_processor, init_load_back）：唯一的变更文件，修复了 match_prefix 中空 key 的边界条件导致 IndexError 的问题。

关键符号：_reset_full, match_prefix, _match_post_processor, init_load_back

关键源码片段

python/sglang/srt/mem_cache/unified_radix_cache.py

唯一的变更文件，修复了 match_prefix 中空 key 的边界条件导致 IndexError 的问题。

```
# python/sglang/srt/mem_cache/unified_radix_cache.py

# 在 _reset_full 方法末尾缓存一个空的 MatchResult，避免重复创建
self._empty_match_result = MatchResult(
    device_indices=torch.empty(
        (0,),
        dtype=torch.int64,
        device=self.device,
    ),
    last_device_node=self.root_node,
    last_host_node=self.root_node,
)

def match_prefix(self, params: MatchPrefixParams) -> MatchResult:
    result = self.session.try_match_prefix(params)
    if result is not None:
        return result

    key = params.key
    key, _ = key.maybe_to_bigram_view(self.is_eagle)
    if self.disable or len(key) == 0:
        # key 为空时直接返回缓存结果，避免进入 _match_prefix_helper
        return self._empty_match_result
    key = key.page_aligned(self.page_size)
    if len(key) == 0:
        # page_aligned 后 key 可能仍然为空，此时也需提前返回
        return self._empty_match_result

    value, last_node, best_value_len = self._match_prefix_helper(key)
    return self._match_post_processor(params, value, last_node, best_value_len)
```

评论区精华

审核者 hzh0425 要求作者检查所有使用了 empty_cache_result 的地方（实际为 _empty_match_result），作者回应已完成。最终审核者指出应与 PR #24470 对齐，暗示有其他 PR 也修改了类似逻辑。

- 统一所有 empty_cache_result 使用位置 (correctness): 作者完成统一，所有原本直接创建空 MatchResult 或空 tensor 的地方都改为使用 self._empty_match_result。
- 与 PR#24470 对齐 (design): 作者确认已完成对齐，但具体对齐内容未在评论中详细说明。

风险与影响

- 风险：风险较低。变更仅影响空 key 的匹配路径，且通过缓存结果避免重复 tensor 创建，不会引入新功能或修改现有逻辑。但需要确认 _empty_match_result 的初始化时机（_reset_full 中）是否在所有可能使用它的地方之前，以及是否考虑了多线程安全（当前 _empty_match_result 仅在初始化时创建一次，后续只读，无竞态问题）。

- 影响：仅影响使用了 UnifiedRadixCache 且 key 为空的情况，典型场景是 HiCache + MTP 组合。修复了调度器崩溃，增强稳定性，对性能几乎无影响（空匹配命中时更快，因为省去了 _match_prefix_helper 调用）。
- 风险标记：边界条件修复，需确认与 PR#24470 对齐

关联脉络

- PR #24470 未明确，但评论中提及：审核者要求本 PR 与 PR#24470 对齐，表明两个 PR 修改了相似的逻辑，可能涉及空匹配结果的优化或重构。