PR #25413 完整报告

sgl-project/sglang

[lora] Fix overlap loading for cancelled requests

合并时间：2026-05-25 14:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25413

执行摘要

- 一句话：修复重叠加载时取消请求导致 LoRA slot 泄漏
- 推荐动作：值得精读。该 PR 展示了一个经典的状态管理 bug 及其优雅的修复方式：用不变式（uid_to_buffer_id）替代可变状态（事件字典）来判断加载完成。对于理解重叠加载的实现和设计 trade-off 很有帮助。

功能与动机

PR body 指出：当调度器实际调度某个 LoRA 的请求时，事件才会从 lora_to_overlap_load_event 中移除。如果该 LoRA 的所有请求在加载完成前被取消，则事件永远不会被清除，导致 LoRA slot 泄漏，甚至瘫痪整个引擎。

实现拆解

1. 在 LoRAOverlapLoader.try_overlap_load_lora 方法开头，新增对 _drain_completed_overlap_loads 的调用，确保在状态检查和容量验证前先清理所有已完成的事件。
2. 重写 _check_overlap_load_status 方法：如果 lora_id 仍在 lora_to_overlap_load_event 字典中，返回 LOADING；否则检查 self.lora_manager.memory_pool.uid_to_buffer_id 中是否存在该 id，存在则返回 LOADED，否则返回 NOT_LOADED。不再在方法内部直接删除事件或调用 event.query()。
3. 新增 _drain_completed_overlap_loads 方法：遍历 lora_to_overlap_load_event，对 event.query() 返回 True 的事件执行 current_stream().wait_event(event) 并删除字典项。
4. 测试文件 test_lora_overlap_loading.py 新增三个单元测试：验证已完成的过期事件在容量检查前被清理、清理后已加载的 LoRA 可被复用、以及 pending 加载必须在 memory pool 有空位时完成。同时增加辅助方法 _mark_loras_loaded 来模拟加载完成时的效果。

关键文件：

- python/sglang/srt/lora/lora_overlap_loader.py（模块 加载器；类别 source；类型 core-logic；符号 _drain_completed_overlap_loads, _check_overlap_load_status）：核心逻辑变更：修改了加载状态判断、新增 drain 方法、调整了事件清理机制
- test/registered/lora/test_lora_overlap_loading.py（模块 测试；类别 test；类型 test-coverage；符号 _mark_loras_loaded, test_completed_stale_loads_are_reaped_before_capacity_check, test_loaded_lora_reused_after_stale_event_drain,

test_pending_lora_load_must_complete_even_if_memory_pool_has_slot）：新增 3 个单元测试覆盖修复逻辑，验证陈旧事件清理、重用和 pending 场景

关键符号：try_overlap_load_lora, _check_overlap_load_status, _drain_completed_overlap_loads, _try_start_overlap_load

关键源码片段

python/sglang/srt/lora/lora_overlap_loader.py

核心逻辑变更：修改了加载状态判断、新增 drain 方法、调整了事件清理机制

```
def _drain_completed_overlap_loads(self) -> None:
    completed_loads = [
        (lora_id, event)
        for lora_id, event in self.lora_to_overlap_load_event.items()
        if event.query() # CUDA 事件查询，True 表示已完成
    ]
    for lora_id, event in completed_loads:
        torch.cuda.current_stream().wait_event(event) # 确保后续操作等待完成
        del self.lora_to_overlap_load_event[lora_id] # 从跟踪字典中移除

def _check_overlap_load_status(self, lora_id: Optional[str]) -> LoRAOverlapLoadStatus:
    # 如果还在事件字典中，说明加载尚未完成（因为已完成的事件已被 drain）
    if lora_id in self.lora_to_overlap_load_event:
        return LoRAOverlapLoadStatus.LOADING
    # 不在事件字典，但 memory pool 中有映射 => 加载已完成
    if lora_id in self.lora_manager.memory_pool.uid_to_buffer_id:
        return LoRAOverlapLoadStatus.LOADED
    # 两者都没有 => 从未开始加载
    return LoRAOverlapLoadStatus.NOT_LOADED
```

test/registered/lora/test_lora_overlap_loading.py

新增 3 个单元测试覆盖修复逻辑，验证陈旧事件清理、重用和 pending 场景

```
def _mark_loras_loaded(self, new_loras, _loras_to_be_loaded):
    # 模拟 fetch_new_loras 完成后的效果：将新加载的 LoRA 加入到 memory pool 的 uid_to_buffer_
    id 映射中
    for lora_id in new_loras:
        self.mock_lora_manager.memory_pool.uid_to_buffer_id[lora_id] = len(
            self.mock_lora_manager.memory_pool.uid_to_buffer_id
        )

def test_completed_stale_loads_are_reaped_before_capacity_check(self):
    # 验证：当有一个陈旧事件已完成时，它在容量检查前被清理，新 LoRA 可以开始加载
    ...
```

评论区精华

Reviewer glenliu21 建议将 _drain_completed_overlap_loads 调用从 _check_overlap_load_status 内移至 try_overlap_load_lora 顶部，以更清楚地体现“尽可能回收已完成事件”的意图。作者 erikwijmans 采纳建议并更新了代码。

- drain 调用位置 (design): 作者 erikwijmans 采纳并更新代码

风险与影响

- 风险：主要风险在于状态判断逻辑的改变：之前事件删除只发生在 _check_overlap_load_status 中，现在统一在 _drain_completed_overlap_loads 中处理，并且在 _check_overlap_load_status 中完全依赖 uid_to_buffer_id。如果 memory pool 的 uid_to_buffer_id 更新时机与事件完成不同步，可能导致加载状态误判。不过由于事件 drain 和状态检查位于同一个 try_overlap_load_lora 调用中，drain 先执行，风险较低。此外，该方法新增对 lora_manager.memory_pool 的访问，若 memory_pool 尚未初始化则可能导致 AttributeError；但测试已覆盖正常流程。
- 影响：修复了 LoRA 重叠加载场景下的 slot 泄漏问题，避免引擎因所有 slot 泄漏而 stall。对于大量使用 LoRA 重叠加载的用户影响显著，稳定性提升。改动范围仅限于 lora_overlap_loader.py 及其测试文件，不影响其他模块。
- 风险标记：核心逻辑变更，依赖 memory pool 状态同步

关联脉络

- 暂无明显关联 PR