PR #31608 完整报告

sgl-project/sglang

[LoRA] Guard TMA down path for LoRA hooks

合并时间：2026-07-22 00:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31608

执行摘要

- 一句话：修复 LoRA 钩子冲突导致 TMA 路径错乱的问题
- 推荐动作：值得精读。该 PR 是一个极佳的 bug 根因分析示例，展示了如何从用户报告追溯到跨多个提交的回归点，并通过最小改动修复。同时，它也揭示了 TMA 布局与路由主序布局不兼容这一架构层面的设计约束，未来在处理类似缓冲区布局问题时值得参考。

功能与动机

修复 Issue #29157 中描述的 CUDA graph 下并发多 LoRA 解码输出乱码（重复的 ! 字符）的问题。该问题根因在于 #23019 重构后，LoRA 钩子意外地进入了专家排序的 TMA 布局，导致钩子读写错误的 token/route 行，包括未初始化的填充行，从而损坏了 LoRA 增量和底层激活值。PR body 中详细追溯了从 #10567 到 #23019 三个提交的回归过程。

实现拆解

1. 定位回归点：在 _fused_moe_kernel_sequence 函数中，down_moe_use_tma 变量控制下投影是否使用 TMA 布局。该变量在 hooks 存在时未做判断，导致 LoRA 钩子安装后仍使用 TMA 路径。
2. 添加防护：在 down_moe_use_tma 被用于计算 padded_tokens 之前，加入条件判断：如果 hooks 不为 None 且 hooks.after_gate_up 或 hooks.after_down 不为 None，则将 down_moe_use_tma 置为 False。
3. 保留非钩子路径：对于没有安装钩子的 MoE 调用，TMA 路径保持不变，完全不影响性能。
4. 测试验证：未引入新的测试文件，但 PR body 中提供了详细的端到端 A/B 测试数据，证明防护后乱码率从 ~90% 降至 0%。评论中 jybsuper 触发了 /rerun-failed-ci 重跑 CI，表明 CI 已通过。

关键文件：

- python/sglang/srt/layers/moe/moe_runner/triton_utils/fused_moe.py（模块 MoE 调度；类别 source；类型 core-logic；符号 _fused_moe_kernel_sequence）：核心修复文件。在 _fused_moe_kernel_sequence 函数中新增 6 行防护代码，当 LoRA 钩子存在时禁用 TMA 下投影路径，恢复历史行为。

关键符号：_fused_moe_kernel_sequence

关键源码片段

python/sglang/srt/layers/moe/moe_runner/triton_utils/fused_moe.py

核心修复文件。在 _fused_moe_kernel_sequence 函数中新增 6 行防护代码，当 LoRA 钩子存在时禁用 TMA 下投影路径，恢复历史行为。

```
# python/sglang/srt/layers/moe/moe_runner/triton_utils/fused_moe.py 第 476-481 行（新增）
    # LoRA 钩子消费和更新按路由主序（route-major）组织的中间缓冲区，
    # 而 TMA 下投影路径将这些缓冲区按专家排序（expert-sorted）并带有块填充存储，
    # 这与钩子契约不兼容。因此，当存在活跃的 after_gate_up 或 after_down 钩子时，
    # 必须禁用 TMA 路径，以避免钩子误读未初始化填充行或错误关联 token/ 专家行。
    if hooks and (hooks.after_gate_up is not None or hooks.after_down is not None):
        down_moe_use_tma = False
```

评论区精华

无审阅评论，但 PR body 本身包含了详尽的根因分析和性能对比。作者论证了禁用 TMA 而非适配布局的理由：TMA 从未在钩子调用中启用过，因此防护只是恢复历史行为；替代方案（保持钩子中间缓冲区路由主序，仅在下 GEMM 边界处聚集为专家排序）虽然原型化并基准测试，但带来了额外的布局转换复杂性，且性能增益（decode -0.09%，prefill +2.26%）在噪声范围内，不值得投入。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。防护仅影响安装了 LoRA 钩子的调用，且仅在底层硬件支持 TMA（如 H200/H20）时才生效。将 down_moe_use_tma 置为 False 后，后续逻辑与无 TMA 路径完全一致，不会引入新的回归。对于没有钩子的调用，代码路径与之前完全相同。唯一的细微风险是如果将来有新的钩子类型加入，需要同步更新此防护条件。
- 影响：对用户：修复了 GLM-5.1-FP8 等模型在启用 CUDA graph 和多 LoRA 时的输出乱码问题，显著提升可靠性。对系统：无性能退化，因为防护仅在钩子安装时生效，且钩子路径原本就不应进入 TMA。对团队：6 行代码、单文件修改，低侵入性，易于维护和理解。
- 风险标记：核心路径变更

关联脉络

- PR #10567 Add TMA down-projection path for MoE (first introduction): 首次引入下投影 TMA 路径，奠定了 down_moe_use_tma 变量和专家排序布局的基础。
- PR #21858 Introduce LoRA MoE hooks in TritonRunnerCore.run: 引入了 LoRA 钩子机制，但当时钩子位于不含 TMA 代码的路径中，因此无条件约束。
- PR #23019 De-duplicate MoE kernel implementations into _fused_moe_kernel_sequence: 将两个实现合并为统一的 _fused_moe_kernel_sequence，但未在钩子存在时禁用 TMA，成为本 PR 修复的回归点。
- PR #29157 [Bug] CUDA graph causes garbled outputs with concurrent multi-LoRA decoding on GLM-5.1-FP8: 用户报告的 bug，直接驱动了本 PR 的修复。