PR #24395 完整报告

sgl-project/sglang

Fix deterministic inference on models with `SWAKVPool`

合并时间：2026-05-05 20:20

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24395

执行摘要

- 一句话：修复 SWAKVPool 模型确定性推理的 CUDA 非法内存访问
- 推荐动作：该 PR 值得精读。它展示了如何诊断由索引空间不一致导致的内存越界问题，并利用缓存设计（swa_loc）优化性能。对于涉及自定义 KV 池（如 SWAKVPool）及统一注意力内核的开发者，此修复具有重要参考价值。

功能与动机

Issue #24394 报告了在 Gemma4 等使用 SWAKVPool 的模型上，启用 --enable-deterministic-inference 和 --attention-backend triton 后，并发请求导致 'CUDA error: an illegal memory access'。问题根源在于 _fwd_kernel_unified 内核读取新 token 时使用了 out_cache_loc（全池索引空间），而 SWAKVPool.set_kv_buffer 将数据写入的是 SWA 转换后的索引空间。两者不一致导致内存越界。本 PR 通过对齐索引空间来修复此问题。

实现拆解

1. 索引空间对齐：在 triton_backend.py 的 _forward_extend_unified 函数中，当检测到当前层为滑动窗口层（sliding_window_size > -1）且使用 SWAKVPool 时，将 extend_kv_indices 从 out_cache_loc（全池索引）转换为 SWA 索引空间。优先使用 pool.swa_loc（预计算缓存），若为空则调用 pool.translate_loc_from_full_to_swa() 进行转换。
2. 缓存优化：swa_loc 由 SWAKVPool 在 set_kv_buffer 时预生成并缓存，避免每次 extend 都重新计算转换，提高效率。
3. 回归测试：新增 test/registered/core/test_gemma4_moe_deterministic.py，模拟 bug 报告的精确环境（Gemma-4-26B-A4B 模型，TP=2，Triton 注意力后端，deterministic 模式，BF16 等），通过 ThreadPoolExecutor 并发 128 发送 200 个请求，并断言全部成功，验证修复有效。

关键文件：

- test/registered/core/test_gemma4_moe_deterministic.py（模块 回归测试；类别 test；类型 test-coverage；符号 TestGemma4MoeDeterministic, setUpClass, tearDownClass, _fire_one）：新增回归测试文件，精确复现 issue #24394 的场景，确保修复有效且未来不退化。

- python/sglang/srt/layers/attention/triton_backend.py（模块 注意力层；类别 source；类型 core-logic；符号 _forward_extend_unified, SWAKVPool, translate_loc_from_full_to_swa, swa_loc）：核心修复文件，在 _forward_extend_unified 方法中添加索引空间转换逻辑，对齐 SWAKVPool 写端和读端的索引空间，消除 CUDA illegal memory access。

关键符号：_forward_extend_unified, test_no_ima_under_concurrent_load

关键源码片段

python/sglang/srt/layers/attention/triton_backend.py

核心修复文件，在 _forward_extend_unified 方法中添加索引空间转换逻辑，对齐 SWAKVPool 写端和读端的索引空间，消除 CUDA illegal memory access。

```
# ... 前置代码：处理滑动窗口参数 ...

# For SWA layers, mirror SWAKVPool.set_kv_buffer: read from the
# precomputed pool.swa_loc. Translate out_cache_loc to SWA-pool index space
# as a fallback when pool.swa_loc is not pre-populated.
extend_kv_indices = forward_batch.out_cache_loc
pool = forward_batch.token_to_kv_pool
if (
    layer.sliding_window_size is not None
    and layer.sliding_window_size > -1
    and isinstance(pool, SWAKVPool)
    and pool.layers_mapping[layer.layer_id][1]
):
    # 优先使用预缓存的 swa_loc，避免重复转换
    if pool.swa_loc is not None:
        extend_kv_indices = pool.swa_loc
    else:
        # 手动转换为 SWA 索引空间，与 set_kv_buffer 写端保持一致
        extend_kv_indices = pool.translate_loc_from_full_to_swa(
            extend_kv_indices
        )

# 后续代码：构建 unified_kv_indices 并使用 extend_kv_indices
```

评论区精华

在 Review 中，ispobock 指出初始注释过于冗余，要求精简，作者随后进行了压缩。同时，ispobock 询问是否可以为该修复添加单元测试，作者最终添加了 test_gemma4_moe_deterministic.py 作为回归测试。所有评论均已解决并通过审批。

- 注释过长 (style): 作者在最终提交（commit 4c4f7de）中缩短了注释。
- 添加单元测试 (testing): 作者添加了测试文件 test/registered/core/test_gemma4_moe_deterministic.py，在回归测试中验证了修复。

风险与影响

- 风险：修复作用域为 *deterministic-extend* 路径，但 SWAKVPool 的其他操作（如 *decode*）是否也存在类似的索引空间不匹配问题？需进一步排查。性能方面，新增的条件分支仅在滑动窗口层且使用 SWAKVPool 时激活，影响范围有限；swa_loc 缓存避免了重复转换开销。测试覆盖仅针对 Gemma4-26B-A4B 模型，其他使用 SWAKVPool 的模型（如 Gemma3）未在本次测试中验证，但修复逻辑具有普适性。
- 影响：用户影响：之前因 CUDA 非法内存访问而无法使用确定性推理的 Gemma 系列模型用户，现在可以正常使用 *--enable-deterministic-inference* 功能，推理结果可复现。系统影响：修复了确定性推理路径中的一个关键 bug，提升了该功能的稳定性和可用性。团队影响：新增的回归测试可在 CI 中持续运行，防止未来退化。
- 风险标记：核心路径变更，并发 / 竞态问题修复，索引空间敏感

关联脉络

- PR #24394 Bug: *--enable-deterministic-inference causes CUDA illegal memory access on models with SWAKVPool under load*: 原始 bug 报告，本 PR 修复该 issue