

PR #33810 完整报告

sgl-project/sglang

fix(gdn): skip the -1 padding sentinel in the chunked extend kernel

合并时间: 2026-08-07 09:48

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33810>

执行摘要

- 一句话: 修复 GDN chunked extend 内核 -1 哨兵越界访问
- 推荐动作: 值得精读。修复展示了 Triton 内核中 `boundary_check` 与真实分配校验的差异, 以及以哨兵值防御 padded 行的通用模式; PR body 给出了严谨的对照实验设计, 是排查 CUDA graph 崩溃问题的范本。

功能与动机

PR body 明确指出: `_forward_metadata` 将 padded 请求行的 `mamba_cache_indices[real_bs:]` 置为 -1 作为哨兵, 解码内核 `fused_recurrent.py` 有 `if idx >= 0` 守卫, 但 chunked gated-delta-net extend 内核直接将该索引用于指针算术, 在状态池之前寻址。`boundary_check` 无法捕获该越界, 因为它校验的是声明块形状而非真实分配。可打断 prefill CUDA graph 与 DP attention 是暴露窗口, 此前依赖 `--disable-prefill-cuda-graph` 规避。

实现拆解

1. 根因定位: `_forward_metadata` 将 padded 请求行的 `mamba_cache_indices` 置为 -1, 作为非法状态索引哨兵; 解码内核 `fused_recurrent.py` 用 `if idx >= 0` 守卫该哨兵, 但 `python/sglang/kernels/ops/attention/fla/chunk_delta_h.py` 中的 `chunk_gated_delta_rule_fwd_kernel_h_blockdim64` 直接将该索引用作指针偏移, 导致在状态池地址之前进行 load/store, 且 `boundary_check` 只校验声明块形状、不校验真实分配, 因此无法拦截。
2. 内核修复: 在函数中新增 `valid_state = index >= 0` 计算, 并将两处访问改为条件执行: `initial-state` 加载改为 `if USE_INITIAL_STATE and valid_state`, `final-state` 就地更新改为 `if INPLACE_UPDATE and valid_state`, 使 padded 行不再执行任何读写。
3. 测试配套: `test/registered/8-gpu-models/test_qwen35.py` 移除 DP 变体中的 `--disable-prefill-cuda-graph` workaround, 让 TP8+DP8 与 TP8+DP8+MTP 重新覆盖可打断 prefill CUDA graph (BCG) 路径; H200 与 B200 上重跑该测试均通过。
4. 验证数据: PR body 提供 4xGB300 上 Qwen3.5-397B-A17B-FP8 (tp4/dp4、DP attention、gsm8k 200q) 对照实验, 修复前 BCG + EAGLE 发生 NCCL AllGather 死锁、无 EAGLE 时发生 `CUDAGraph::replay SIGSEGV`; 修复后两种配置均通过, EAGLE 配置精度 0.965、吞吐 1780 tok/s。

关键文件：

- `python/sglang/kernels/ops/attention/fla/chunk_delta_h.py`（模块 扩展内核；类别 source；类型 core-logic；符号 `chunk_gated_delta_rule_fwd_kernel_h_blockdim64`）：核心修复文件：在 chunked GDN extend 内核中跳过 -1 哨兵行，防止状态池越界访问。
- `test/registered/8-gpu-models/test_qwen35.py`（模块 集成测试；类别 test；类型 test-coverage）：测试配套：移除 DP 变体的 `--disable-prefill-cuda-graph` 规避参数，重新覆盖可打断 prefill CUDA graph 路径。

关键符号：`chunk_gated_delta_rule_fwd_kernel_h_blockdim64`

关键源码片段

`python/sglang/kernels/ops/attention/fla/chunk_delta_h.py`

核心修复文件：在 chunked GDN extend 内核中跳过 -1 哨兵行，防止状态池越界访问。

```
# chunk_delta_h.py 中 chunk_gated_delta_rule_fwd_kernel_h_blockdim64 的关键修复段
index = tl.load(initial_state_indices + i_n).to(tl.int64)
# Padded rows carry the -1 sentinel; the decode kernel guards on it
# (fused_recurrent.py), the chunked extend path did not.
# 填充行携带 -1 哨兵；解码内核有守卫，但 chunked extend 路径之前没有。
valid_state = index >= 0
h0 = initial_state + index * stride_init_state
ht = initial_state + index * stride_init_state

# load initial state
# 仅对有效行加载初始状态，防止访问状态池之前的地址
if USE_INITIAL_STATE and valid_state:
    p_h0_1 = tl.make_block_ptr(h0, (V, K), (K, 1), (i_v * BV, 0), (BV, 64), (1, 0))
    b_h1 += tl.load(p_h0_1, boundary_check=(0, 1)).to(tl.float32)
    # 若 K > 64，此处还有第二个 K 分块的 load，逻辑相同

# epilogue
# 仅对有效行写回最终状态，避免越界 store
if INPLACE_UPDATE and valid_state:
    p_ht = tl.make_block_ptr(ht, (V, K), (K, 1), (i_v * BV, 0), (BV, 64), (1, 0))
    tl.store(p_ht, b_h1.to(p_ht.dtype.element_ty), boundary_check=(0, 1))
    # 若 K > 64，此处还有第二个 K 分块的 store，逻辑相同
```

评论区精华

Review 无实质讨论，Fridge003 直接批准合并。Issue 评论中作者与 reviewer 各发起一次 [/rerun-test test/registered/8-gpu-models/test_qwen35.py](#)，最终 H200 与 B200 均通过。PR body 中的对照实验是最有价值的设计证据。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 修改限定在 `chunk_delta_h.py` 单个内核，`decode` 路径不受影响。
 - 仅跳过加载与回写，`padded` 行的中间计算仍可能产生无效数据，但不会写回状态池；且这些行本就应被视为无效。
 - 防御逻辑依赖哨兵值 `-1` 的语义，若未来其他调用方传入非 `-1` 的非法索引（如 `-2`），仍可能越界，建议后续补充内核级单元测试锁定该契约。
 - 测试仅覆盖 Qwen3.5 模型配置，未覆盖其他使用 GDN 扩展内核的模型，存在回归盲区。
- 影响：
 - 对用户：修复了在 8 卡 H200/B200 上运行 Qwen3.5-397B 配合 DP attention 与可打断 CUDA graph 时的死锁或崩溃。
 - 对系统：DP arm 可恢复使用 `prefill` CUDA graph，EAGLE 场景实测吞吐约 1780 tok/s、精度 0.965。
 - 对团队：改动量小、风险可控，但需注意内核行为与 `_forward_metadata` 哨兵约定的耦合。
 - 风险标记：核心内核路径变更，哨兵约定耦合，缺少内核级单元测试，验证覆盖单一模型

关联脉络

- PR #33666 fix(PP): size the mamba pool per pipeline stage, not per whole model: 同为 mamba/GDN 状态池相关修复，涉及状态池容量与索引语义，是本 PR 的领域邻居。
- PR #33794 Fix paged SWA retraction resume accounting: 同为状态缓存索引与记账问题，且均涉及 `padded` 行与调度路径，可对照理解越界根因模式。