PR #34561 完整报告

sgl-project/sglang

[Fix] Fix Nemotron-H Mamba illegal memory access under DP attention with CUDA graph

合并时间：2026-08-20 05:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34561

执行摘要

- 一句话：修复 DP attention 下 Nemotron-H Mamba CUDA graph 崩溃
- 推荐动作：值得精读，因为它展示了 DP attention 与 CUDA graph 后端交互时的关键坑点及修复模式，尤其对从事模型推理内核和 CUDA graph 优化的工程师有参考价值。关注点：如何将 *eager* 路径与 CUDA graph 路径统一，以及 fuse_mlp_allreduce 在 DP 下的特殊处理。

功能与动机

运行 Nemotron-H 时，启用 DP attention 并配合 CUDA-graph runner 后端（breakable CUDA graph / torch.compile piecewise）会在图回放时崩溃。PR body 中明确说明：'Running Nemotron-H with DP attention enabled together with a CUDA-graph runner backend ... crashes at graph replay'，并附有 torch.AcceleratorError: CUDA error: an illegal memory access was encountered 的堆栈。根本原因是 DP 分支绕过了 CUDA graph 所需的分段执行逻辑。

实现拆解

1. 修改核心 forward 逻辑：在 python/sglang/srt/models/nemotron_h.py 的 NemotronHMambaDecoderLayer.forward 中，is_dp_attention_enabled() 分支原先直接调用 self._forward_mamba(hidden_states, forward_batch)，现改为与 CUDA graph 相关路径相同的 split-op 分发。变更后：is_in_breakable_cuda_graph() 时调用 breakable_nemotron_mamba2_with_output(hidden_states, output, self.layer_id, False)；is_in_tc_piecewise_cuda_graph() 时调用 nemotron_mamba2_with_output(hidden_states, output, self.layer_id, False)；否则保持 *eager* 路径调用 self._forward_mamba。这样保证了 CUDA graph 捕获和回放时能走正确分段逻辑。
2. 显式设置 fuse_mlp_allreduce=False：DP 路径不计算 fuse_mlp_allreduce（allreduce 由 layer communicator 处理），因此此处显式传 False，与非 DP 路径的行为保持一致，避免在 DP 下错误融合。
3. 调整测试配置以覆盖场景：在 test/registered/4-gpu-models/test_nvidia_nemotron_3_super_nvfp4.py 的 DP_ATTENTION_EP_ARGS 中移除了 --cuda-graph-backend-prefill 和 disabled 两个参数。之前显式禁用 prefill 的 CUDA graph，导致测试无法覆盖此崩溃；移除后测试默认启用 breakable CUDA graph 后端，从而能够验证修复。
4. 验证：PR body 中给出修复后的 gsm8k 结果为 0.9432，且 4-gpu-b200 的测试通过。

关键文件：

- python/sglang/srt/models/nemotron_h.py（模块 模型层；类别 source；类型 core-logic；符号 NemotronHMambaDecoderLayer.forward）：核心修复文件，修改了 DP attention 分支的 forward 逻辑，使其复用 CUDA graph 的分段执行路径，解决了崩溃问题。
- test/registered/4-gpu-models/test_nvidia_nemotron_3_super_nvfp4.py（模块 测试；类别 test；类型 test-coverage；符号 DP_ATTENTION_EP_ARGS）：调整测试配置，移除禁用 CUDA graph 的参数，使测试覆盖到本修复的场景。

关键符号：NemotronHMambaDecoderLayer.forward, breakable_nemotron_mamba2_with_output, nemotron_mamba2_with_output

关键源码片段

python/sglang/srt/models/nemotron_h.py

核心修复文件，修改了 DP attention 分支的 forward 逻辑，使其复用 CUDA graph 的分段执行路径，解决了崩溃问题。

```
# python/sglang/srt/models/nemotron_h.py
# NemotronHMambaDecoderLayer.forward 的 DP attention 分支
# 修复前：直接调用 _forward_mamba，绕过了 CUDA graph 分段执行
# 修复后：根据 CUDA graph 后端类型分发，避免图回放时的非法内存访问
if is_dp_attention_enabled():
    hidden_states, residual = self._dp_attn_input(hidden_states, residual, forward_batch)
    if get_real_num_tokens(hidden_states, forward_batch) == 0:
        return torch.zeros_like(hidden_states), residual

    # 根据 CUDA graph 模式选择正确的执行路径
    if is_in_breakable_cuda_graph():
        output = torch.empty_like(hidden_states)
        # 使用 breakable CUDA graph 的 Mamba2 执行函数
        breakable_nemotron_mamba2_with_output(hidden_states, output, self.layer_id, False)
    elif is_in_tc_piecewise_cuda_graph():
        output = torch.empty_like(hidden_states)
        # 使用 torch.compile piecewise 的 Mamba2 执行函数
        nemotron_mamba2_with_output(hidden_states, output, self.layer_id, False)
    else:
        # eager 路径不变
        output = self._forward_mamba(hidden_states, forward_batch)
    return output, residual
```

test/registered/4-gpu-models/test_nvidia_nemotron_3_super_nvfp4.py

调整测试配置，移除禁用 CUDA graph 的参数，使测试覆盖到本修复的场景。

```
# test/registered/4-gpu-models/test_nvidia_nemotron_3_super_nvfp4.py
# 移除禁用 CUDA graph 的参数，使测试覆盖到 DP attention + breakable CUDA graph 的崩溃场景
DP_ATTENTION_EP_ARGS = [
    "--dp-size", "4",
```

```
    "--enable-dp-attention",
    "--enable-dp-lm-head",
    "--ep-size", "4",
    "--moe-a2a-backend", "flashinfer",
    "--moe-runner-backend", "flashinfer_cutedsl",
    "--mamba-full-memory-ratio", "5.0",
    "--mamba-radix-cache-strategy", "extra_buffer",
    "--attention-backend", "trtllm_mha",
    "--max-running-requests", "1024",
    "--mem-fraction-static", "0.93",
    "--max-prefill-tokens", "8192",
    # 原先这里加了 "--cuda-graph-backend-prefill", "disabled"，现已移除
]
```

评论区精华

review 过程中，b8zhong（Nemotron 相关维护者）触发了 /rerun-test test/registered/4-gpu-models/test_nvidia_nemotron_3_super_nvfp4.py，并最终APPROVE。mmangkad 也 APPROVE。评论线程中 nvpohanh 提到 *cc @b8zhong since this is related to Nemotron*，表明该修复与 Nemotron 模型相关。没有发现对设计方案的质疑或未解决的讨论。

- 测试重跑与验收 (testing): 测试通过，PR 获得 APPROVE。
- Nemotron 相关性确认 (question): b8zhong 确认并参与 review，最终 approve。

风险与影响

- 风险：
  1. 回归风险：修改直接影响 NemotronHMambaDecoderLayer.forward 的 DP attention 分支，影响所有启用 DP attention 的 Nemotron-H 推理。但非 DP 路径未改动，*eager* 路径也保持原逻辑，因此回归面有限。
  2. CUDA graph 兼容性：修复依赖 is_in_breakable_cuda_graph() 和 is_in_tc_piecewise_cuda_graph() 的准确判断；若这些函数在 DP 场景下有误判，可能仍会崩溃。不过测试已覆盖该场景。
  3. 性能影响：拆分为 breakable 或 piecewise 执行可能带来轻微开销，但这是 CUDA graph 后端的正常路径，对性能影响应可接受。
  4. 测试覆盖：删除 --cuda-graph-backend-prefill disabled 后，测试默认启用 breakable CUDA graph，但测试运行在 4-GPU 环境，且 *coverage* 可能不完整（如其他 CUDA graph 后端）。
- 影响：
  1. 对用户：修复了 Nemotron-H 在 DP attention + CUDA graph 场景下的崩溃，使该配置可用，提升了稳定性和可用性。
  2. 对系统：变更仅影响 Nemotron-H 模型的 DP attention 路径，对其他模型无影响。
  3. 对团队：为 Nemotron-H 的 DP + CUDA graph 组合提供了可复用的模式，后续类似模型可参考该修复方式。

4. 影响程度：中等，属于特定模型特定配置的 bugfix，但修复了导致崩溃的关键路径。 - 风险标记：核心路径变更，缺少单测覆盖，依赖 CUDA graph 模式判断

关联脉络

- PR #35269 [UnifiedTree] feat: support runtime attach/detach: 涉及 Mamba 相关内存管理，可能与本 PR 的 Mamba DP 路径有交互，但非直接相关。
- PR #35545 [Qwen3.5][MTP] Preserve online NVFP4 draft quantization for mixed checkpoints: 同为 NVFP4 量化模型修复，但不同模型，关联性低。