PR #34935 完整报告

sgl-project/sglang

[NPU]Ensure tensors allocated by empty_like are contiguous

合并时间：2026-08-20 20:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34935

执行摘要

- 一句话：修复 NPU 注意力输出非连续布局问题
- 推荐动作：值得精读，虽然改动简单，但体现了对 PyTorch 内存格式继承行为的深刻理解，可作为类似防御性修复的参考。建议后续考虑补充针对非连续输入 / 输出的单元测试，以强化回归保障。

功能与动机

PR 描述指出 torch.empty_like(q) 默认继承 q 的内存格式，当 q 非连续时，attn_output 也会非连续，可能引发下游 NPU 算子对连续张量的要求导致失败。作者在 Motivation 中明确说明了这一风险，旨在增强 NPU 推理的鲁棒性。

实现拆解

1. 定位分配点：在 python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py 的 forward_extend（约 1590 行）和 forward_decode（约 2749 行）中，找到 torch.empty_like(q) 的调用。
2. 修改分配方式：在两处调用中均添加 memory_format=torch.contiguous_format 参数，确保分配的张量内存连续。该改动不影响 qk_head_dim != v_head_dim 分支（使用 q.new_empty），因为该分支已显式指定形状，默认为连续。
3. 配套改动：无测试文件变更，但 CI 运行了 NPU 相关测试（标签 run-ci），并通过了 accuracy 测试（gemma-3-4b-it）。

关键文件：

- python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py（模块 NPU 注意力；类别 source；类型 core-logic）：包含 NPU 注意力前向路径的 *extend* 和 *decode* 核心逻辑，本次修改直接提升输出张量的连续性，是修复的核心文件。

关键符号：forward_extend, forward_decode, run_sdpa_forward_extend

关键源码片段

python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py

包含 NPU 注意力前向路径的 *extend* 和 *decode* 核心逻辑，本次修改直接提升输出张量的连续性，是修复的核心文件。

```
# 位于 python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py
# forward_extend 与 forward_decode 中构造 attn_output 的部分

if layer.qk_head_dim != layer.v_head_dim:
    # qk 与 v 维度不同时，用 q.new_empty 分配明确形状的连续张量
    attn_output = q.new_empty((q.shape[0], layer.tp_q_head_num * layer.v_head_dim))
else:
    # 关键修改：显式指定 contiguous_format，避免继承 q 的非连续内存布局
    # 这确保了后续 view 和 NPU 算子操作的安全性
    attn_output = torch.empty_like(q, memory_format=torch.contiguous_format)

# 后续使用 view 重新组织 head 维度，连续布局是 view 能正确工作的前提
use_gqa = layer.tp_q_head_num != layer.tp_k_head_num
q_ = q.view(-1, layer.tp_q_head_num, layer.qk_head_dim)
o_ = attn_output.view(-1, layer.tp_q_head_num, layer.v_head_dim)
```

评论区精华

Review 过程中无具体讨论内容，仅由 sglang-npu-bot 自动批准，PR 作者通过评论触发 CI 重跑。整体上属于低争议的防御性修复。

- 暂无高价值评论线程

风险与影响

- 风险：改动位于注意力核心路径，影响 NPU 上所有模型的注意力输出分配。显式指定连续内存格式可能带来轻微的内存复制开销，但通常远小于因非连续导致的算子失败；风险较低。由于没有新增针对性单元测试，对非连续输入场景的回归覆盖不足，但 CI 的 accuracy 测试提供了基本保障。
- 影响：影响范围限于 NPU 硬件后端，不会影响其他硬件（如 CUDA）。影响程度为正向改进，能减少 NPU 推理中因内存布局引发的偶发错误，提升稳定性。对用户无感知，对团队而言是一次低成本高收益的防御性修复。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #35197 fix(kernel) Fix Helion small-token prefill bug: 同为注意力内核缺陷修复，涉及张量形状与维度处理，属于同类 bugfix 模式。