PR #23819 完整报告

sgl-project/sglang

[NPU] Fix warmup error with --disable-cuda-graph and mtp

合并时间：2026-05-11 09:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23819

执行摘要

- 一句话：NPU MTP warmup 因 padding token 维度不匹配崩溃修复
- 推荐动作：值得合入，修复明确且验证充分。review 中的建议（使用 forward_batch.batch_size）可作为后续优化参考，但不影响当前正确性。

功能与动机

PR body 指出服务器 warmup 时启用了 --disable-cuda-graph 后，forward_decode 中 torch.ops.npu.npu_fused_infer_attention_score 会因输入张量中的 padding token 导致维度不匹配错误。

实现拆解

1. 裁剪 query 张量：在 python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py 的 forward_decode 函数中，首先保存原始的 token 总数 num_token_padding = q.shape[0]，然后从 block_tables 获取实际的 batch 大小 actual_bs，并通过 q = q[:actual_bs] 截取前 actual_bs 个 token。
2. 调用 NPU 融合注意力算子：使用裁剪后的 q 调用 torch.ops.npu.npu_fused_infer_attention_score，注意 q.view 的维度从 (forward_batch.batch_size, -1, ...) 改为 (-1, 1, ...)，因为此时 q 的形状已经是 (actual_bs, ...)。
3. 恢复原始形状：如果裁剪后的 token 数不等于原始 padded token 数，则通过 torch.cat 将算子输出的 attn_output 与零张量拼接，恢复为 (num_token_padding, ...) 形状，确保下游逻辑不受影响。
4. 仅影响 NPU FIA 路径：该修改仅在 self.use_fia 为 True 时生效，不影响其他注意力后端（如 torch_npu._npu_paged_attention 或 alibi 路径）。

关键文件：

- python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py（模块 NPU 后端；类别 source；类型 core-logic）：NPU 注意力后端核心文件，修复 MTP warmup 时 query 张量 padding 导致维度不匹配的 bug。

关键符号：未识别

关键源码片段

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

NPU 注意力后端核心文件，修复 MTP warmup 时 query 张量 padding 导致维度不匹配的 bug。

```
# python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py
# forward_decode 方法中，当使用 NPU 融合注意力算子时（self.use_fia 为 True）：

if self.use_fia:
    if self.forward_metadata.seq_lens_cpu_int is None:
        actual_seq_len_kv = self.forward_metadata.seq_lens_cpu_list
    else:
        actual_seq_len_kv = (
            self.forward_metadata.seq_lens_cpu_int.cpu().int().tolist()
        )
    # 保存原始 token 总数（含 padding）
    num_token_padding = q.shape[0]
    # 从 block_tables 获取实际 batch 大小（即真实序列数）
    actual_bs = self.forward_metadata.block_tables.shape[0]
    # 裁剪 query：只保留前 actual_bs 个 token，移除 MTP 引入的冗余 padding
    q = q[:actual_bs]
    attn_output, _ = torch.ops.npu.npu_fused_infer_attention_score(
        q.view(
            -1,
            1, # 注意：此处从 (batch_size, -1) 改为 (-1, 1)，因为 q 已裁剪为 (actual_bs, ...)
            layer.tp_q_head_num,
            layer.qk_head_dim,
        ),
        k_cache.view(...),
        v_cache.view(...),
        num_heads=layer.tp_q_head_num,
        num_key_value_heads=layer.tp_k_head_num,
        input_layout="BSND",
        atten_mask=None,
        block_size=self.page_size,
        block_table=self.forward_metadata.block_tables,
        actual_seq_lengths_kv=actual_seq_len_kv,
        scale=layer.scaling,
    )
    # 如果裁剪过，则将输出用零填充回原始形状，保证后续逻辑不受影响
    if actual_bs != num_token_padding:
        attn_output = torch.cat(
            [
                attn_output,
                attn_output.new_zeros(
                    num_token_padding - actual_bs,
                    *attn_output.shape[1:],
                ),
            ],
```

```
            dim=0,
        )
```

评论区精华

review 中 gemini-code-assist[bot] 建议使用 forward_batch.batch_size 替代从 block_tables 推导 actual_bs，以保持代码一致性并避免 forward_metadata 未完全初始化时的风险。但作者未采纳该建议，最终合并版本仍使用 self.forward_metadata.block_tables.shape[0]。

- 使用 forward_batch.batch_size 替代 block_tables 推导实际 batch 大小 (design): 作者未采纳，最终使用 self.forward_metadata.block_tables.shape[0]，PR 已合入。

风险与影响

- 风险：
  - 回归风险低：修改仅限于 if self.use_fia: 分支，且添加了裁剪 / 填充逻辑，不会影响其他注意力路径。若 actual_bs 计算错误（例如 block_tables 形状不反映真实 batch 大小），可能导致裁剪过多或过少，但 MTP warmup 场景下已验证一致。
  - 性能影响可忽略：仅在 MTP warmup 时少量增加裁剪和拼接操作，对稳态推理无影响。
  - 未添加单元测试：PR 未包含对应测试，但提供了 ceval 精度验证。
- 影响：
  - 用户影响：修复了 NPU 上使用 MTP + --disable-cuda-graph 时的 server warmup 崩溃，使该配置组合可用。
  - 系统影响：对系统性能无负面影响。
  - 团队影响：变更极小（+15/-1），合入风险低。
  - 风险标记：核心路径变更

关联脉络

- 暂无明显关联 PR