PR #26705 完整报告

sgl-project/sglang

[Bugfix] Fix Ascend NPU CP attention for batch size > 1

合并时间：2026-05-30 15:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26705

执行摘要

- 一句话：修复 Ascend NPU CP 注意力 bs>1 崩溃
- 推荐动作：建议合并，修复明确且经过 review 验证。值得关注的设计决策是：CP 泛化后 NPU 路径的遗漏修复方式 —— 使用 total_q_prev_tokens 作为 Q 分割点而非全局二分。

功能与动机

PR #23269 将 ContextParallelMetadata 的标量字段改为每序列的列表 / 张量，以支持 bs > 1 的注意力 CP。但 Ascend NPU 的 do_cp_attn_fia 方法未更新，仍使用已删除的标量字段 actual_seq_q_prev、kv_len_prev 等，导致 AttributeError；同时 Q 分割使用 torch.chunk(q, 2) 仅对 bs==1 正确，bs>1 时 zigzag 布局要求按 total_q_prev_tokens 分割。该修复使 Qwen3-MoE 等非 MLA 模型在 NPU 上能端到端运行 CP。

实现拆解

1. Q 分割修正（ascend_backend.py:831-836）：将 torch.chunk(q, 2, dim=0) 改为按 cp_meta.total_q_prev_tokens 索引切片 q[:split] 和 q[split:]，确保 zigzag 布局下 prev/next 半段正确对应各序列的前 / 后块。
2. 序列长度参数更新（ascend_backend.py:858-859,875-876）：将 actual_seq_lengths=[cp_meta.actual_seq_q_prev] 替换为 np.cumsum(cp_meta.actual_seq_q_prev_list).tolist()，将标量引用改为每序列累计和列表，与新的 ContextParallelMetadata 格式对齐；actual_seq_lengths_kv 同理从 [cp_meta.kv_len_prev] 改为 cp_meta.kv_len_prev_list。
3. 配套调整：根据 review 建议，将 np.cumsum 结果通过 .tolist() 转换为 Python 列表，避免 NPU 自定义算子类型不匹配（第二 commit）。

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；符号 do_cp_attn_fia）：唯一修改文件，修复 Ascend NPU FIA CP 路径的 Q 分割和序列长度参数，使其支持 bs>1。

关键符号：do_cp_attn_fia

关键源码片段

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

唯一修改文件，修复 Ascend NPU FIA CP 路径的 Q 分割和序列长度参数，使其支持 bs>1。

```
  def do_cp_attn_fia(
      self,
      q: torch.Tensor,
      k_cache: torch.Tensor,
      v_cache: torch.Tensor,
      layer: "RadixAttention",
      forward_batch: ForwardBatch,
  ) -> torch.Tensor:
      """CP-aware attention for non-MLA models using FIA on Ascend NPU."""
      cp_meta = forward_batch.attn_cp_metadata

      # Local tokens are laid out [all_seqs_prev, all_seqs_next]; split at
      # total_q_prev_tokens rather than the midpoint to support bs > 1.
      # torch.chunk(q, 2) was only correct for bs == 1.
      split = cp_meta.total_q_prev_tokens
      q_prev = (
          q[:split].contiguous().reshape(-1, layer.tp_q_head_num, layer.qk_head_dim)
      )
      q_next = (
          q[split:].contiguous().reshape(-1, layer.tp_q_head_num, layer.qk_head_dim)
      )

      k_cache_paged = k_cache.view(
          -1, self.page_size, layer.tp_k_head_num * layer.qk_head_dim
      )
      v_cache_paged = v_cache.view(
          -1, self.page_size, layer.tp_v_head_num * layer.v_head_dim
      )

      attn_out_prev, _ = torch.ops.npu.npu_fused_infer_attention_score(
          q_prev,
          k_cache_paged,
          v_cache_paged,
          block_table=self.forward_metadata.block_tables,
          block_size=self.page_size,
          num_heads=layer.tp_q_head_num,
          num_key_value_heads=layer.tp_k_head_num,
          input_layout="TND",
          atten_mask=self.fia_mask,
          sparse_mode=3,
          next_tokens=0,
          scale=layer.scaling,
          # Use cumulative per-seq lengths (list of ints) instead of removed scalar fields.
          actual_seq_lengths=np.cumsum(cp_meta.actual_seq_q_prev_list).tolist(),
          actual_seq_lengths_kv=cp_meta.kv_len_prev_list,
      )
```

```
    attn_out_next, _ = torch.ops.npu.npu_fused_infer_attention_score(
        q_next,
        k_cache_paged,
        v_cache_paged,
        block_table=self.forward_metadata.block_tables,
        block_size=self.page_size,
        num_heads=layer.tp_q_head_num,
        num_key_value_heads=layer.tp_k_head_num,
        input_layout="TND",
        atten_mask=self.fia_mask,
        sparse_mode=3,
        next_tokens=0,
        scale=layer.scaling,
        actual_seq_lengths=np.cumsum(cp_meta.actual_seq_q_next_list).tolist(),
        actual_seq_lengths_kv=cp_meta.kv_len_next_list,
    )

    attn_out = torch.cat([attn_out_prev, attn_out_next], dim=0)
    return attn_out.view(-1, layer.tp_q_head_num * layer.v_head_dim)
```

评论区精华

Review 中 gemini-code-assist[bot] 指出两次 np.cumsum 调用返回 NumPy 数组，而 NPU 自定义算子通常需要 Python list，建议添加 .tolist() 转换。该建议被采纳并在第二 commit 中实现。无其他争议。

- np.cumsum 返回类型需转换为 Python list (correctness): 作者接受建议，在第二 commit 中添加 .tolist() 转换。

风险与影响

- 风险：回归风险低，仅修改 NPU FIA CP 路径，不影响 CUDA、MLA、DSA/NSA 路径。bs==1 时 np.cumsum([x]) == [x] 且 total_q_prev_tokens == actual_seq_q_prev，向后兼容。主要风险是 NPU 算子 npu_fused_infer_attention_score 对 cumulative lengths 格式可能有特殊约束（已在 review 中处理）。
- 影响：影响范围限于 Ascend NPU 上使用 --enable-prefill-context-parallel 且 ASCEND_USE_FIA=1 的非 MLA 模型（如 Qwen3-30B-A3B）。修复前 CP 完全不可用（AttributeError），修复后 bs>1 可正常工作。用户无需修改配置。团队维护成本低，代码量仅 +13/-10。
- 风险标记：缺少测试覆盖，NPU 专用路径

关联脉络

- PR #23269 Generalize context-parallel prefill from bs==1 to bs>1: 本 PR 修复了 #23269 泛化 CP 时遗漏的 NPU FIA 路径。