# PR #21685 完整报告

sgl-project/sglang

[NPU] ascend backend support qwen3 moe attention cp

合并时间：2026-04-29 19:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21685

## 执行摘要

- 一句话：Ascend NPU 为 Qwen3 MoE 标准注意力添加 CP
- 推荐动作：建议阅读 _cp_allgather_and_save_kv_npu 的合并通信策略以及 do_cp_attn_fia 的 zigzag 实现，这对类似 CP 实现有参考价值。测试设计也值得学习。

## 功能与动机

Qwen3 MoE 模型在 Ascend NPU 上已支持 MLA 注意力路径的 Prefill Context Parallel (PCP)，但标准注意力路径缺少 CP 支持。本 PR 填补这一空白，使 Qwen3-30B-A3B 等非 MLA 模型也能在 co-located 部署中利用 CP 降低长序列 prefill 的 HBM 占用，改善 TTFT。

## 实现拆解

1. 新增 _cp_allgather_and_save_kv_npu 函数（ascend_backend.py）：将 K 和 V 展平后沿特征维度拼接，通过一次 cp_all_gather_rerange_kv_cache 完成跨秩通信，再拆解回 K/V 缓存。对 GQA 场景（tp_k_head_num != tp_v_head_num）同样有效。
2. 新增 do_cp_attn_fia 方法（ascend_backend.py）：实现 CP 感知的 Attention 计算。根据 attn_cp_size 和 cp_rank，将 Q 按 zigzag 模式拆分为前一半和后一半，分别调用 npu_fused_infer_attention_score 计算，最后拼接结果输出。
3. 修改 forward_extend 方法（ascend_backend.py）：当 is_context_parallel_extend 为 True 时，先执行 all-gather KV，再调用 do_cp_attn_fia 代替常规 FIA。若非 FIA 路径（如 NZ 格式）则抛出 NotImplementedError。
4. 存储 attn_cp_size：在 __init__ 中从 model_runner.attn_cp_size 读取并保存到 self.attn_cp_size。
5. 测试覆盖：新增 test_npu_qwen3_30b_attn_cp.py，注册为 nightly-4-npu-a3 套件。使用 TP=4 / MOE_DP=2 / ATTN_CP=2 启动服务器，在 100 条 GSM8K 样本上验证准确率 ≥ 0.92。
6. 文档更新：在 ascend_npu_qwen3_examples.md 中添加 Qwen3-235B-A22B 的 PCP 配置示例，包含 Prefill 和 Decode 节点参数说明。

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；符号 _cp_allgather_and_save_kv_npu, do_cp_attn_fia）：核心实现文件，添加 CP KV all-gather 和 CP FIA 注意力方法

- test/registered/ascend/llm_models/test_npu_qwen3_30b_attn_cp.py（模块 集成测试；类别 test；类型 test-coverage；符号 TestQwen330BAttnCP, setUpClass, tearDownClass, test_gsm8k_accuracy）：新增 nightly CI 测试，验证 GSM8K 准确率
- docs/platforms/ascend/ascend_npu_qwen3_examples.md（模块 文档；类别 docs；类型 documentation）：添加 Qwen3-235B-A22B 的 PCP 配置示例

关键符号：_cp_allgather_and_save_kv_npu, do_cp_attn_fia

## 关键源码片段

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

核心实现文件，添加 CP KV all-gather 和 CP FIA 注意力方法

```
def _cp_allgather_and_save_kv_npu(forward_batch, layer, k, v, cp_size):
    """NPU 兼容的 CP KV all-gather，合并 K/V 通信.

    将 K 和 V 沿特征维度拼接，只需一次 all-gather 而非两次，减少一半通信延迟。

    k shape: [S_local, tp_k_head_num, qk_head_dim]
    v shape: [S_local, tp_v_head_num, v_head_dim]

    等价于 cp_utils.py 中的 cp_allgather_and_save_kv_cache()，但使用一次 all-gather。
    """
    cache_loc = (
        forward_batch.out_cache_loc
        if not layer.is_cross_attention
        else forward_batch.encoder_out_cache_loc
    )
    # 保存原始尾部形状，用于 all-gather 后 reshape
    k_tail = k.shape[1:] # (tp_k_head_num, qk_head_dim)
    v_tail = v.shape[1:] # (tp_v_head_num, v_head_dim)

    # 展平尾部维度然后拼接 — 一次 all-gather 而非两次
    # 对 GQA 也适用，即使 tp_k_head_num != tp_v_head_num
    k_flat = k.contiguous().reshape(k.shape[0], -1) # [S_local, k_feat]
    v_flat = v.contiguous().reshape(v.shape[0], -1) # [S_local, v_feat]
    k_feat_size = k_flat.shape[-1]
    kv_flat = torch.cat([k_flat, v_flat], dim=-1) # [S_local, k_feat + v_feat]

    kv_full = cp_all_gather_rerange_kv_cache(
        kv_flat, cp_size, forward_batch, get_current_device_stream_fast()
    ) # [S_full, k_feat + v_feat]

    key_cache_full = kv_full[..., :k_feat_size].reshape(-1, *k_tail)
    value_cache_full = kv_full[..., k_feat_size:].reshape(-1, *v_tail)

    forward_batch.token_to_kv_pool.set_kv_buffer(
        layer,
        cache_loc,
```

```
        key_cache_full,
        value_cache_full,
    )
```

## 评论区精华

主要 review 讨论：

- 设备流正确性：gemini-code-assist[bot] 指出 _cp_allgather_and_save_kv_npu 中使用 torch.cuda.current_stream() 在 NPU 上不正确。最终代码使用 get_current_device_stream_fast() 解决。
- 代码重复：审查者建议将重复的 FIA 调用提取为 helper。作者 AndyLi429 认为不必要（回复“unnecessary”），未修改。
- 设备流正确性：使用 torch.cuda.current_stream 的风险 (correctness): 最终代码使用 get_current_device_stream_fast() 统一处理，问题解决。
- 代码重复：建议提取 FIA 调用为 helper (design): 作者 AndyLi429 认为不必要（回复“unnecessary”），未修改。

## 风险与影响

- 风险：
  - 回归风险：NPU 后端注意力路径被修改，可能影响其他 NPU 模型。但只有 CP 分支影响，非 CP 路径不变。
  - 性能风险：合并 all-gather 减少了通信，但增加了拼接和拆解开销。实测性能提升 13%。
  - 兼容性风险：FIA 路径依赖 ASCEND_USE_FIA=1，若未设置环境变量则 CP 路径抛出 NotImplementedError，用户明确得知不支持。
  - 测试覆盖：只有 GSM8K 端到端测试，缺乏单元测试覆盖边界情况（如单 token prefill、不同 CP 大小）。
- 影响：
  - 用户：使用 Ascend NPU + Qwen3 MoE（非 MLA）的用户可以利用 CP 降低长序列 prefill 的峰值显存，改善 TTFT。需要设置 --attn-cp-size 和 --enable-prefill-context-parallel。
  - 系统：影响范围限定在 NPU 后端的 co-located 部署模式；其他后端的计算不受影响。
  - 团队：代码增加约 300 行（核心逻辑 + 测试 + 文档），维护成本较低。
  - 风险标记：NPU 后端核心变更，CP 路径依赖 FIA，非 FIA 路径显式 UNSUPPORTED

## 关联脉络

- 暂无明显关联 PR