PR #20177 完整报告

sgl-project/sglang

[NPU]adapt multibatch fia ops

合并时间：2026-05-11 09:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/20177

执行摘要

- 一句话：NPU FIA 算子多批处理优化，替代逐序列循环
- 推荐动作：建议阅读该 PR，了解 NPU 注意力优化如何利用 CANN 多 batch 特性。关注 fia_mask 硬编码问题，后续可能需要提 PR 修复。设计决策——用 TND 布局替换 BSND 并处理 padding——值得参考。

功能与动机

之前，逐序列 for 循环影响性能。现在，FIA 操作支持多 batch 输入。

实现拆解

1. 移除逐序列 for 循环：原来对每个序列调用 npu_fused_infer_attention_score 的 BSND 输入，现在改为单次调用。
2. 采用 TND 布局批量计算：通过传入 k_cache/v_cache 视图、block_table、block_size、实际序列长度等参数，支持变长序列的多 batch 注意力计算。
3. 处理 padding token：若 q 的 token 数量超过实际非 padding token 数（prefix cache 场景），则先截断 q/k/v，再在计算结果后补零对齐。
4. 扩展适用模型：将使用条件由 self.use_fia 扩展到 self.use_fia or layer.qk_head_dim == 256，覆盖 qwen3-next 等模型。
5. 配置键调整：同步更新注释和配置键字段。这些改动集中在 python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py 的 forward_extend 方法中。

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 forward_extend）：唯一变更文件，核心注意力逻辑重构，性能提升关键。

关键符号：forward_extend

关键源码片段

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

唯一变更文件，核心注意力逻辑重构，性能提升关键。

```
if self.use_fia:
    """FIA supports multi-bs in the current version of CANN"""
    q = q.reshape(-1, layer.tp_q_head_num, layer.qk_head_dim)
    num_token_padding = q.shape[0]
    # 如果存在 padding token（prefix cache 场景），截断 q/k/v 到实际 token 数
    if num_token_padding > forward_batch.num_token_non_padded_cpu:
        q, k, v = [
            data[: forward_batch.num_token_non_padded_cpu]
            for data in [q, k, v]
        ]
    # 单次调用 npu_fused_infer_attention_score，传入 k_cache/v_cache 视图和 block_table
    attn_output, _ = torch_npu.npu_fused_infer_attention_score(
        query=q,
        key=k_cache.view(-1, self.page_size, layer.tp_k_head_num * layer.qk_head_dim),
        value=v_cache.view(-1, self.page_size, layer.tp_v_head_num * layer.v_head_dim),
        block_table=self.forward_metadata.block_tables,
        block_size=self.page_size,
        atten_mask=self.fia_mask,
        input_layout="TND",
        actual_seq_lengths=self.forward_metadata.seq_lens_list_cumsum,
        actual_seq_lengths_kv=self.forward_metadata.seq_lens_cpu_int,
        num_key_value_heads=layer.tp_k_head_num,
        num_heads=layer.tp_q_head_num,
        scale=layer.scaling,
        sparse_mode=3,
    )
    attn_output = attn_output.view(-1, layer.tp_q_head_num * layer.v_head_dim)
    # 如果之前有 padding token，需要在输出后面补零保持形状一致
    if num_token_padding != forward_batch.num_token_non_padded_cpu:
        attn_output = torch.cat(
            [
                attn_output,
                attn_output.new_zeros(
                    num_token_padding - attn_output.shape[0],
                    *attn_output.shape[1:],
                ),
            ],
            dim=0,
        )
```

评论区精华

AI 代码审查机器人 gemini-code-assist 指出硬编码的 fia_mask 最大长度 2048 可能导致越界和 DoS 风险，尤其是对长上下文模型如 qwen3-next，但该问题在合并时未解决。开发者 zhuyijie88 建议添加条件判断来处理 num_token_padding 大于真实 token 数时的切片逻辑，该建议已被采纳。

- 硬编码 fia_mask 2048 越界风险 (security): PR 合并时未处理该问题，后续需修复。

- 切片条件处理 padding token (correctness): PR 已采纳该建议，添加了 if 条件进行切片处理。

风险与影响

- 风险：
  - 硬编码 fia_mask 2048：可能导致长上下文模型注意力计算越界，引发 DoS 崩溃，需后续动态扩展。
  - k_cache/v_cache 视图依赖 page 结构和 block_table 正确性，若初始化错误可能访问无效内存。
  - 切片和补零逻辑可能引入精度偏差或 OOM，尤其在 padding 很多时。
  - 改动集中在 NPU 后端核心路径，回归测试依赖于 CI 覆盖。
  - 影响：仅影响 Ascend NPU 后端，当启用 FIA 或模型 qk_head_dim==256（如 qwen3-next）时会走新路径。性能提升约 23%（22us->17us）在给定用例，且支持 prefix cache，对用户透明。需注意模型上下文长度若超过 2048 需额外适配。
  - 风险标记：fia_mask 硬编码 2048 越界风险，NPU 后端注意力核心路径变更

关联脉络

- 暂无明显关联 PR