PR #47198 完整报告

vllm-project/vllm

[Perf] Remove redundant op for GLM 5.2

合并时间：2026-07-05 01:25

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47198

执行摘要

- 一句话：消除 GLM 5.2 稀疏注意力中冗余算子
- 推荐动作：值得合并。改动小、无风险，且是 Issue #46654 性能优化工作中清除冗余算子的典型步骤。对维护者而言，这类零碎优化在长期可积累显著的端到端加速。

功能与动机

作为 Issue #46654（GLM 5.2 性能优化）的一部分，本 PR 旨在移除稀疏注意力计算中的冗余算子，减少小张量操作开销。PR body 明确说明“All changes identical to previous implementation”，确保数值一致性。

实现拆解

1. 预计算缩放因子：在 __init__ 中添加 self.n_head_scale = self.n_head**-0.5，避免每次 forward 调用时重复计算指数和除法。该值在推理过程中恒定。
2. 消除 k_pe 冗余维度操作：在 fused_indexer_q 分支中，将原先的 k_pe.unsqueeze(1)，k_pe.unsqueeze(1) 传给 rotary_emb 后 reshape(-1, 1, self.rope_dim) 再 squeeze(-2) 合并为：先 unsqueeze(1) 传给 rotary_emb，再 reshape(-1, self.rope_dim) 直接得到 MQA 形状 [num_tokens, rope_dim]。类似地，在 else 分支中，直接 reshape(-1, self.rope_dim) 并拼接，省去中间 1 维度操作。
3. 简化 q_scale 维度：将 q_scale.view(-1, self.n_head, 1) 改为 q_scale.view(-1, self.n_head)，因为后续 weights 计算中会做乘法，广播机制能自动处理最后一维。
4. 合并 weights 的乘法和维度操作：原先的 weights.unsqueeze(-1) * q_scale * self.softmax_scale * self.n_head**-0.5 再 squeeze(-1) 改为直接 weights * q_scale * self.softmax_scale * self.n_head_scale，利用广播消除中间张量操作。

关键文件：

- vllm/model_executor/models/deepseek_v2.py（模块 模型执行；类别 source；类型 core-logic；符号 DeepseekAttention.init, DeepseekAttention.forward）：唯一修改的文件，包含 DeepseekAttention 类的优化，涉及稀疏注意力前向传播中多个冗余操作的移除。

关键符号：DeepseekAttention.init, DeepseekAttention.forward

关键源码片段

vllm/model_executor/models/deepseek_v2.py

唯一修改的文件，包含 DeepseekAttention 类的优化，涉及稀疏注意力前向传播中多个冗余操作的移除。

```
class DeepseekAttention(nn.Module):
    def __init__(self, ...):
        ...
        self.is_inplace_rope = is_inplace_rope
        # 预计算缩放因子，避免每次前向都重复计算 self.n_head ** -0.5
        self.n_head_scale = self.n_head ** -0.5
        self.use_fused_indexer_q = (
            current_platform.is_cuda()
            and self.quant_block_size == self.head_dim
            and self.head_dim == 128
            and self.rope_dim == 64
            and self.scale_fmt is not None
        )

    def forward(self, ...):
        ...
        # fused_indexer_q 分支：消除 k_pe 上额外的 unsqueeze/squeeze 操作
        if current_platform.is_rocm() and self.is_inplace_rope:
            ...
        elif self.use_fused_indexer_q and q.dtype == torch.bfloat16:
            ...
            # 原实现是多次 unsqueeze 再 reshape，改为更直接的 reshape
            k_pe = k_pe.unsqueeze(1) # 仍保持 3D 供 rotary_emb 消费
            q_dummy = torch.empty_like(k_pe)
            _, k_pe = rotary_emb(positions, q_dummy, k_pe)
            k_pe = k_pe.reshape(-1, self.rope_dim) # 直接展平最后一维
            k = torch.cat([k_pe, k_nope], dim=-1) # 拼接得到 [num_tokens, head_dim]
            return self.indexer_op(hidden_states, q_fp8, k, weights)
        else:
            ...
            q_pe, k_pe = rotary_emb(positions, q_pe, k_pe.unsqueeze(1))
            q_pe = q_pe.reshape(-1, self.n_head, self.rope_dim)
            k_pe = k_pe.reshape(-1, self.rope_dim) # 直接展平，省去中间 1 维度
            q = torch.cat([q_pe, q_nope], dim=-1)
            k = torch.cat([k_pe, k_nope], dim=-1) # 拼接结果与之前 squeeze 后一致
        # 量化分支：简化 q_scale 的 shape，利用广播机制
        q_fp8, q_scale = per_token_group_quant_fp8(q, ...)
        q_fp8 = q_fp8.view(-1, self.n_head, self.head_dim)
        q_scale = q_scale.view(-1, self.n_head) # 去掉 expand 到 head_dim 维的 view
        # 合并 weights 的乘法和 squeeze，直接使用预计算的 n_head_scale
        weights = weights * q_scale * self.softmax_scale * self.n_head_scale
        return self.indexer_op(hidden_states, q_fp8, k, weights)
```

评论区精华

- 暂无高价值评论线程

风险与影响

- 风险：本 PR 只调整了张量维度操作和预计算常量，没有修改算法逻辑或控制流，数值等价性在 PR body 中声明。主要风险是 fused_indexer_q 分支中 k_pe 的维度操作变化可能影响后续 rotary_emb 的调用，因为 rotary_emb 可能依赖输入形状。但作者通过精简 unsqueeze 次数并保持 rotary_emb 调用时 k_pe 仍为 3D，确保了兼容性。此外，else 分支中去掉了 squeeze(-2)，直接拼 [num_tokens, rope_dim] 的 k_pe，需要确认 k_nope 的形状也为 [num_tokens, head_dim-rope_dim]（保持不变），拼接结果维度正确。整体风险较低。
- 影响：本 PR 是纯性能优化，不影响功能或接口。目标用户是使用 GLM 5.2 模型（DeepseekV2 架构变体）且启用稀疏 MLP 注意力（use_fused_indexer_q）的用户。预计推理速度提升微小但稳定，因为去除了几个小张量操作。由于改动仅在一处文件中，影响范围限于 DeepseekAttention 类。
- 风险标记：暂无

关联脉络

- PR #46654 [Feature]: GLM 5.2 Performance Optimization: 本 PR 是该性能优化跟踪 Issue 的一部分，旨在提升 GLM 5.2 模型的推理性能。
- PR #46862 [Perf] Remove redundant ops on GLM5.2: 同一作者在类似主题上的另一优化 PR，可能涉及同一代码路径。
- PR #46642 [Perf] Remove redundant noop copy in GLM5.2: 同一 Issue 下的另一个性能优化 PR，涉及 GLM 5.2 模型。