PR #26309 完整报告

sgl-project/sglang

[npu] [bugfix] Add contiguous operation during quantized weight loading.

合并时间：2026-05-27 19:55

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26309

执行摘要

- 一句话：修复 NPU 量化 MoE 权重加载非连续导致 EPLB 失败
- 推荐动作：建议阅读此 PR 了解 NPU 量化 MoE 权重加载的潜在陷阱。本 PR 的核心决策（在 transpose 后添加 contiguous()）是解决非连续张量问题的标准做法，值得作为类似问题的参考模式。同时关注关联 PR #25663 可能对同一问题做更彻底的架构演进。

功能与动机

启用 EPLB 后触发权重重新分布，量化模型权重非连续导致传输错误。如 PR body 所述：'Once EPLB is enabled, a weight redistribution process is triggered; this subsequently leads to the following error because the quantized model's weights are non-contiguous.'

实现拆解

1. 定位问题：在 fused_moe_method_npu.py 的 NPUW4A4Int4DynamicMoEMethod 和 NPUW8A8Int8DynamicMoEMethod 的 process_weights_after_loading 方法中，权重 w13_weight 和 w2_weight 经 transpose(1, 2) 后直接传入 npu_format_cast，未保证内存连续。
2. 修改 NPUW4A4Int4DynamicMoEMethod：对 transposed 张量添加 .contiguous()，确保 npu_format_cast 接收连续输入。
3. 修改 NPUW8A8Int8DynamicMoEMethod：同样添加 .contiguous()，保持双路径一致。
4. 验证：使用 Qwen3-30B-A3B-W8A8（NPUW8A8Int8DynamicMoEMethod），TP=2，EPLB 启用下测试，显存无增长，GSM8K 准确率 0.889。
5. 代码质量：应维护者要求修复了 lint 问题。

关键文件：

- python/sglang/srt/hardware_backend/npu/quantization/fused_moe_method_npu.py（模块 量化方法；类别 source；类型 core-logic；符号 NPUW4A4Int4DynamicMoEMethod.process_weights_after_loading, NPUW8A8Int8DynamicMoEMethod.process_weights_after_loading）：PR 唯一修改的文件，在 NPUW4A4Int4DynamicMoEMethod 和 NPUW8A8Int8DynamicMoEMethod 的 process_weights_after_loading 方法中添加 contiguous 调用，修复 EPLB 下量化权重非连续问题。

关键符号：NPUW4A4Int4DynamicMoEMethod.process_weights_after_loading，NPUW8A8Int8DynamicMoEMethod.process_weights_after_loading

关键源码片段

python/sglang/srt/hardware_backend/npu/quantization/fused_moe_method_npu.py

PR 唯一修改的文件，在 NPUW4A4Int4DynamicMoEMethod 和 NPUW8A8Int8DynamicMoEMethod 的 process_weights_after_loading 方法中添加 contiguous 调用，修复 EPLB 下量化权重非连续问题。

```
class NPUW8A8Int8DynamicMoEMethod(_NPUFusedMoEMethodBase):
    def process_weights_after_loading(self, layer: torch.nn.Module) -> None:
        if self._maybe_apply_fuseep_weights(layer):
            return
        # 确保 w13_weight 在 transpose 后内存连续，避免 npu_format_cast 因非连续张量出错
        layer.w13_weight.data = npu_format_cast(
            layer.w13_weight.data.transpose(1, 2).contiguous()
        )
        # 同理对 w2_weight 进行连续化
        layer.w2_weight.data = npu_format_cast(
            layer.w2_weight.data.transpose(1, 2).contiguous()
        )
        # 后续 scale 与 offset 处理与原逻辑一致
        layer.w13_weight_scale = torch.nn.Parameter(
            layer.w13_weight_scale.data.squeeze(-1), requires_grad=False
        )
        layer.w2_weight_scale = torch.nn.Parameter(
            layer.w2_weight_scale.data.squeeze(-1), requires_grad=False
        )
        layer.w13_weight_scale_bf16 = torch.nn.Parameter(
            layer.w13_weight_scale.data.to(dtype=torch.bfloat16), requires_grad=False
        )
        layer.w2_weight_scale_bf16 = torch.nn.Parameter(
            layer.w2_weight_scale.data.to(dtype=torch.bfloat16), requires_grad=False
        )
        # Compressed-tensors 格式可能没有 offset 字段
        if hasattr(layer, "w13_weight_offset"):
            layer.w13_weight_offset = torch.nn.Parameter(
                layer.w13_weight_offset.data.squeeze(-1),
                requires_grad=False,
            )
        if hasattr(layer, "w2_weight_offset"):
            layer.w2_weight_offset = torch.nn.Parameter(
                layer.w2_weight_offset.data.squeeze(-1),
                requires_grad=False,
            )
        # dispatcher 配置省略，保持不变
```

评论区精华

- 审阅者 OrangeRedeng 首先询问 NPUW8A8Int8DynamicMoEMethod 是否也受影响，并担心 .contiguous() 可能导致内存增加。作者用 Qwen3-30B-A3B-W8A8 测试确认问题存在且内存无增长。
- OrangeRedeng 进一步表示自己的 MoE 重构 PR #25663 已包含 .contiguous()，本 PR 可作为临时方案先行合并，长期由重构覆盖。
- 另一维护者 ping1jing2 要求修复 lint 问题，作者已完成。
- NPUW8A8Int8DynamicMoEMethod 是否也有问题及内存影响 (question): 作者用 Qwen3-30B-A3B-W8A8 测试确认 W8A8 同样需要 contiguous，且内存无增长。
- 临时解决方案与长期 MoE 重构的关系 (design): 同意合并，长期由重构覆盖。
- 修复 lint 问题 (style): 作者已修复。

风险与影响

- 风险：
  - 回归风险低：contiguous() 是 PyTorch 原生操作，仅确保内存连续，不会改变数值，不会破坏已有逻辑。
  - 显存影响小：作者测试显示显存无增长。连续化可能引起一次额外拷贝，但对加载路径而言是单次操作，无运行时性能影响。
  - 缺少单元测试覆盖：PR 未新增测试文件，仅靠手动验证。虽然改动简单，但在未来重构中可能引入相同问题。
  - 覆盖范围有限：仅修复了 NPUW4A4Int4DynamicMoEMethod 和 NPUW8A8Int8DynamicMoEMethod，其他量化方法可能未覆盖。
- 影响：
  - 用户影响：所有使用 NPU 后端且启用 EPLB 的量化模型用户（W4A4/W8A8）从此可以正常工作。
  - 系统影响：无性能退化，改动仅作用于权重加载阶段，对 inference 路径无影响。
  - 团队影响：小范围修复，快速合并。与 MoE 重构 PR #25663 形成短期方案与长期方案衔接。
  - 风险标记：缺少单元测试覆盖 , NPU 量化加载路径

关联脉络

- PR #25663 [NPU] MoE refactoring (add contiguous): OrangeRedeng 的 MoE 重构 PR 已包含 .contiguous()，是此问题的长期解决方案。