PR #52002 完整报告

vllm-project/vllm

[Bugfix] compressed-tensors: restore int8 grouped WNA16 MoE support

合并时间：2026-08-19 22:56

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52002

执行摘要

- 一句话：恢复 int8 分组 WNA16 MoE 支持
- 推荐动作：此 PR 值得精读，因为它涉及一个关键的回归修复，并揭示了重构中容易丢失的小修复。建议关注其缺失的测试覆盖，并考虑为类似场景添加回归测试。

功能与动机

在 #44570 的重构中，将 CompressedTensorsWNA16MarlinMoEMethod 合并到 CompressedTensorsWNA16MoEMethod 时，保留了后者原有的 assert self.group_size == -1，导致先前 #47154 修复的功能丢失。这导致任何专家未统一量化宽度的压缩张量检查点无法加载，自 v0.27.0 起出现回归。该 PR 旨在恢复修复，确保 int8 分组量化（如 group_size=128）的 MoE 模型能正常加载。

实现拆解

1. 修改入口：在 vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/compressed_tensors_moe_wna16.py 的 CompressedTensorsWNA16MoEMethod.__init__ 中，删除 num_bits == 8 分支的 assert self.group_size == -1 断言（第 85 行）。
2. 核心逻辑：删除断言后，int8 分支的 scale 仍然设置为 kInt8StaticGroupScale，且后续的 weight_key 构建和 select_wna16_moe_backend 调用不受影响，允许 group_size 为任意值（如 128），从而支持分组量化。
3. 配套测试：未添加测试文件，PR 描述中提供了重现方法与测试计划，但未包含自动化测试。

关键文件：

- vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/compressed_tensors_moe_wna16.py（模块 量化方法；类别 source；类型 data-contract；符号 CompressedTensorsWNA16MoEMethod.init）：核心变更文件，删除 assert self.group_size == -1 以恢复 int8 分组 WNA16 MoE 支持。

关键符号：CompressedTensorsWNA16MoEMethod.init

关键源码片段

vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/compressed_tensors_moe_wna16.py

核心变更文件，删除 assert self.group_size == -1 以恢复 int8 分组 WNA16 MoE 支持。

```
# vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/
compressed_tensors_moe_wna16.py
class CompressedTensorsWNA16MoEMethod:
    def __init__(
        self,
        weight_quant: QuantizationArgs,
        input_quant: QuantizationArgs | None,
        moe: FusedMoEConfig,
        layer_name: str | None = None,
    ):
        super().__init__(moe)
        self.weight_quant = weight_quant
        ...
        # 根据量化位宽选择 scale 类型
        if self.num_bits == 4:
            if self.group_size == 32:
                scale = kInt4Static32GroupScale
            else:
                scale = kInt4StaticGroupScale
        elif self.num_bits == 8:
            # 移除原先的 assert self.group_size == -1,
            # 以支持 int8 分组量化（如 group_size=128）的 MoE 专家加载。
            scale = kInt8StaticGroupScale
        else:
            raise ValueError(
                "CompressedTensorsWNA16MoEMethod only supports int4 and int8 now."
            )
        ...
```

评论区精华

Review 评论较少，无实质技术讨论。mgoin 批准了 PR。可能的关注点在于缺少测试覆盖，但 PR 提供了清晰的复现方法。

- 暂无高价值评论线程

风险与影响

- 风险：删除断言后，int8 分组 WNA16 MoE 得以加载，但需确认后端（如 Marlin）确实支持该配置。根据 #47154 的描述，check_moe_marlin_supports_layer 已允许 group_size 为 -1、32、64、128，因此风险较低。但该变更未新增测试，可能引入未覆盖的回归。此外，断言删除后，其他不支持分组 int8 的后端可能被错误选择，但当前代码通过 select_wna16_moe_backend 进行校验。

- 影响：影响范围限于压缩张量库的 MoE 量化路径，特别是 int8 分组量化。修复后，混合精度专家（如 int4/int8 混合）的模型可正常加载，扩大了对 poolside/Laguna-XS 等模型的支持。对现有功能影响较小，因为仅删除了一个断言，未改变其他逻辑。
- 风险标记：缺少测试覆盖 , 回归风险低

关联脉络

- PR #47154 [Bugfix] compressed-tensors: allow int8 grouped WNA16 MoE on Marlin: 原始修复 PR，本 PR 恢复其变更。
- PR #44570 [MoE Refactor] Combine CompressedTensorsWNA16MarlinMoEMethod with CompressedTensorsWNA16MoEMethod: 重构 PR，导致本次回归。