PR #41161 完整报告

vllm-project/vllm

Fix static actorder handling for compressed-tensors WNA16 MoE

合并时间：2026-06-23 06:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41161

执行摘要

- 一句话：修复 static actorder 下 WNA16 MoE w2 scales sharding
- 推荐动作：值得精读的设计决策：将 sharding 规则提取为纯静态方法，方便单元测试和后续扩展；早失败原则（对不可整除 size 抛出异常）优于运行时 CUDA 崩溃。建议未来关注枚举替换字符串的后续 PR。

功能与动机

当使用 actorder=static（等效于 weight-order，无运行时 g_idx）的 compressed-tensors WNA16 MoE checkpoint 时，tp=2, group_size=32 的 Qwen3-A3B W4A16 模型在加载时因 Marlin kernel 推断出无效 group size 而崩溃。根本原因是原有逻辑将任何 truthy actorder 值视为 grouped actorder，导致 w2 scales 未按 TP 分区 shard 且 is_k_full 为 False。

实现拆解

1. 提取 sharding 决策逻辑：在 compressed_tensors_moe_wna16_marlin.py 中新增静态方法 _w2_scale_sharding，根据 actorder 是否为 "group" 决定是否加载完整 w2 scales（load_full_w2）以及 is_k_full 的值。
2. 改进 create_weights 调用：在 create_weights 中调用 _w2_scale_sharding 替代原有内联逻辑，并移除旧注释。
3. 添加整除性验证：当 load_full_w2=False 时，检查 intermediate_size_per_partition % group_size == 0，否则抛出清晰的 ValueError。
4. 同步到非 Marlin 后端：在 compressed_tensors_moe_wna16.py 中添加相同的整除性验证，确保 Flashinfer 等后端也获得早期错误反馈。
5. 单元测试覆盖：新增参数化测试 test_wna16_marlin_moe_w2_scale_sharding，覆盖 "group"、"static"、"weight"、None 及 channel-wise 场景，验证 _w2_scale_sharding 返回值符合预期。

关键文件：

- vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/compressed_tensors_moe_wna16_marlin.py（模块 量化层；类别 source；类型 core-logic；符号 _w2_scale_sharding, create_weights）：核心变更文件，新增 _w2_scale_sharding 方法和整除性验证，修复 static actorder 下的 w2 scale sharding 逻辑。

- vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/compressed_tensors_moe_wna16.py（模块 量化层；类别 source；类型 core-logic；符号 create_weights）：补充添加相同的分区可整除性验证，确保非 Marlin 后端（如 Flashinfer）也能及早发现不兼容的 TP 配置。
- tests/quantization/test_compressed_tensors.py（模块 测试；类别 test；类型 test-coverage；符号 test_wna16_marlin_moe_w2_scale_sharding）：新增参数化单元测试，全面覆盖各种 actorder/group_size/TP 分区场景，确保修复正确性并防止回归。

关键符号：CompressedTensorsWNA16MarlinMoEMethod._w2_scale_sharding, CompressedTensorsWNA16MarlinMoEMethod.create_weights, CompressedTensorsWNA16MoEMethod.create_weights

关键源码片段

vllm/model_executor/layers/quantization/compressed_tensors/compressed_tensors_moe/compressed_tensors_moe_wna16_marlin.py

核心变更文件，新增 _w2_scale_sharding 方法和整除性验证，修复 static actorder 下的 w2 scale sharding 逻辑。

```
@staticmethod
def _w2_scale_sharding(
    actorder,
    group_size: int,
    intermediate_size_per_partition: int,
    intermediate_size_full: int,
) -> tuple[bool, int, bool]:
    """Decide how to shard w2 group scales across TP for WNA16 Marlin MoE.

    Only ``actorder="group"`` permutes activations by ``g_idx`` at runtime
    and therefore needs the full-K (unsharded) w2 scales plus ``is_k_full``.
    ``actorder="weight"``/``"static"`` (and ``None``) reorder weights at
    quantization time, so scales shard normally per TP rank.
    """
    # 仅当 actorder == "group" 且 group_size != -1 时才需要完整加载 w2 scales
    load_full_w2 = (actorder == "group") and group_size != -1
    w2_scales_size = (
        intermediate_size_full if load_full_w2 else intermediate_size_per_partition
    )
    # is_k_full 在非 group 或分区等于完整大小时为 True
    is_k_full = (actorder != "group") or (
        intermediate_size_per_partition == intermediate_size_full
    )
    return load_full_w2, w2_scales_size, is_k_full
```

tests/quantization/test_compressed_tensors.py

新增参数化单元测试，全面覆盖各种 actorder/group_size/TP 分区场景，确保修复正确性并防止回归。

```
@pytest.mark.parametrize(
    "actorder,group_size,part,full,expected",
    [
        # actorder="group" with real grouping: must load full-K w2 scales and,
        # when sharded (part != full), report is_k_full=False.
        (ActivationOrdering.GROUP, 32, 64, 128, (True, 128, False)),
        # actorder="group" but unsharded (part == full): full scales, k_full.
        (ActivationOrdering.GROUP, 32, 128, 128, (True, 128, True)),
        # actorder="group" with channel-wise (group_size == -1): no full load.
        (ActivationOrdering.GROUP, -1, 64, 128, (False, 64, False)),
        # "static"/"weight" reorder at quant time -> shard normally + k_full.
        # Regression: static actorder under TP must keep is_k_full=True so the
        # Marlin kernel never gets the invalid (group_size=16, is_k_full=0).
        ("static", 32, 64, 128, (False, 64, True)),
        ("weight", 32, 64, 128, (False, 64, True)),
        (None, 32, 64, 128, (False, 64, True)),
    ],
)
def test_wna16_marlin_moe_w2_scale_sharding(actorder, group_size, part, full, expected):
    from vllm.model_executor.layers.quantization.compressed_tensors.\
        compressed_tensors_moe.compressed_tensors_moe_wna16_marlin import \
        CompressedTensorsWNA16MarlinMoEMethod
    result = CompressedTensorsWNA16MarlinMoEMethod._w2_scale_sharding(
        actorder, group_size, part, full
    )
    assert result == expected
```

评论区精华

枚举比较 vs 字符串比较：gemini-code-assist[bot] 建议在 _w2_scale_sharding 中使用 ActivationOrdering.GROUP 枚举成员替代硬编码字符串 "group"，以增强健壮性和可维护性。作者 ZewenShen-Cohere 回复同意该建议，但认为应在独立 PR 中统一修改，避免本 PR 范围过大。该讨论未在本 PR 中修改，但已记录为待改进项。

- 使用 ActivationOrdering 枚举替代字符串比较 (design): 作者认为该建议合理，但字符串比较在代码中多处使用，决定在独立 PR 中统一修改，不在本 PR 中调整。

风险与影响

- 风险：类型假设风险：_w2_scale_sharding 中直接比较 actorder == "group"，依赖 compressed-tensors 库将 ActivationOrdering 实现为 StrEnum。若未来库改变实现或用户传入其他类型，可能无法正确匹配。当前风险可控，但建议逐步迁移到枚举比较。兼容性风险：新增的整除性验证可能阻止一些原本能勉强运行但 scale 跨边界的 TP 配置，但这种“失败”比后期 CUDA kernel 崩溃更可取，属于安全增强。
- 影响：用户影响：使用 compressed-tensors WNA16 MoE 且 actorder=static/weight 的用户现在可以正确加载模型并使用 TP>1。修复后 gsm8k 评测 TP=1 与 TP=2 分数接近，精度无损。系统影响：仅影响 compressed_tensors 量化中 MoE 权重 sharding 路径，不影

响其他量化或线性层。团队影响：明确了不同 actorder 类型的语义，提升了代码可读性和可测试性，降低了未来维护成本。

- 风险标记：核心路径变更，枚举比较风险

关联脉络

- 暂无明显关联 PR