PR #44735 完整报告

vllm-project/vllm

[Bugfix] Canonicalize FP8 weight layout to (K, N) at the source

合并时间：2026-06-09 04:37

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44735

执行摘要

- 一句话：修复 FP8 Marlin 方形权重损坏 bug
- 推荐动作：建议精读并 cherry-pick：修复了关键数据损坏 bug，且设计上强化了布局契约，为后续规范化铺路。

功能与动机

修复 #44110：MarlinFP8 核在方形矩阵上产生静默数据损坏，根源是 Marlin 核内部的形状启发式 `if w_q.shape != (layer.input_size_per_partition, layer.output_size_per_partition)` 在 N==K 时始终为 False，跳过转置。替代方案 #44113 使用 `is_contiguous()` 不够健壮。

实现拆解

1. 删除 Marlin 核的布局检测：在 vllm/model_executor/kernels/linear/scaled_mm/marlin.py 的 process_weights_after_loading 中，删除非 block 分支中基于形状的转置条件（19 行），改为注释声明调用方必须传入 (K, N) 布局。
2. Fp8LinearMethod 调用方主动转置：在 vllm/model_executor/layers/quantization/fp8.py 的 process_weights_after_loading 中，对非 block 的 Marlin 路径直接调用 replace_parameter(layer, "weight", layer.weight.t())；对在线量化路径统一在量化后转置为 (K, N)。
3. CompressedTensors 调用方主动转置：在 vllm/model_executor/layers/quantization/compressed_tensors/schemes/compressed_tensors_w8a16_fp8.py 的 process_weights_after_loading 非 block 分支中新增 replace_parameter(layer, "weight", layer.weight.t())。
4. Fp8OnlineLinearMethod 简化：合并 Marlin/ 非 Marlin 分支为统一的转置 + 委托流程，消除重复代码。

关键文件：

- vllm/model_executor/kernels/linear/scaled_mm/marlin.py（模块 量化核；类别 source；类型 data-contract；符号 process_weights_after_loading）：删除 Marlin 核内部的布局检测逻辑，改为调用方负责规范化的契约
- vllm/model_executor/layers/quantization/fp8.py（模块 量化层；类别 source；类型 data-contract；符号 process_weights_after_loading）：在 Fp8LinearMethod 和 Fp8OnlineLinearMethod 中统一添加转置逻辑

- vllm/model_executor/layers/quantization/compressed_tensors/schemes/compressed_tensors_w8a16_fp8.py（模块 量化层；类别 source；类型 data-contract；符号 process_weights_after_loading）：CompressedTensors 路径新增转置，完成全 FP8 路径统一

关键符号：process_weights_after_loading

关键源码片段

vllm/model_executor/kernels/linear/scaled_mm/marlin.py

删除 Marlin 核内部的布局检测逻辑，改为调用方负责规范化的契约

```
# vllm/model_executor/kernels/linear/scaled_mm/marlin.py#L70-L84
    def process_weights_after_loading(self, layer: torch.nn.Module) -> None:
        if self.block_quant:
            weight, weight_scale_inv = process_fp8_weight_block_strategy(
                layer.weight, layer.weight_scale_inv
            )
            replace_parameter(layer, "weight", weight.data)
            replace_parameter(layer, "weight_scale_inv", weight_scale_inv.data)
        # 非 block 分支：调用方必须将权重布局规范化为 (K, N) 再传入
        # 不再由核内部猜测布局，消除方形矩阵时 N==K 导致转置被静默跳过的问题
        layer.input_scale = None
        prepare_fp8_layer_for_marlin(
            layer, self.size_k_first, input_dtype=self.marlin_input_dtype
        )
        del layer.input_scale
```

vllm/model_executor/layers/quantization/fp8.py

在 Fp8LinearMethod 和 Fp8OnlineLinearMethod 中统一添加转置逻辑

```
# vllm/model_executor/layers/quantization/fp8.py#L385-L394 (Fp8MoEMethod)
defprocess_weights_after_loading(self, layer: RoutedExperts) -> None:          if
self.use_marlin:              if not self.block_quant:                  # 将权重规范化为 (K, N) 再
传给 Marlin 核                  replace_parameter(layer, "weight", layer.weight.t())
if hasattr(self.fp8_linear, "marlin_input_dtype"):
self.fp8_linear.marlin_input_dtype = self.marlin_input_dtype
self.fp8_linear.process_weights_after_loading(layer)              return          # ... #
vllm/model_executor/layers/quantization/fp8.py#L533-L554 (Fp8OnlineLinearMethod)
defprocess_weights_after_loading(self, layer: torch.nn.Module) -> None:          # ...
  qweight, weight_scale = ops.scaled_fp8_quant(layer.weight, scale=None)          # 量化
后立即转置为 (K, N)，统一 Marlin 和非 Marlin 路径          replace_parameter(layer, "
weight", qweight.t().data)          replace_parameter(layer, "weight_scale",
weight_scale.data)          if self.use_marlin and hasattr(self.fp8_linear, "
marlin_input_dtype"):              self.fp8_linear.marlin_input_dtype =
self.marlin_input_dtype          self.fp8_linear.process_weights_after_loading(layer)
layer._already_called_process_weights_after_loading = True
```

vllm/model_executor/layers/quantization/compressed_tensors/schemes/compressed_tensors_w8a16_fp8.py

CompressedTensors 路径新增转置，完成全 FP8 路径统一

```
# vllm/model_executor/layers/quantization/compressed_tensors/schemes/compressed_tensors_w8a16_fp8.py#L140-L141
        else:
            # ...
            # 在传入 Marlin 核之前，将权重转置为 (K, N)
            replace_parameter(layer, "weight", layer.weight.t())
        self.linear_kernel.process_weights_after_loading(layer)
```

评论区精华

未产生 review 讨论，已由 robertgshaw2-redhat 批准。

- 暂无高价值评论线程

风险与影响

- 风险：回归风险：所有非 block 的 FP8 Marlin 路径均已验证方形和非方形形状。但 block 量化路径未受影响。若未来添加新的调用方且未转置，将直接暴露。
- 影响：影响所有使用 FP8 Marlin 量化（sm_75-sm_88 GPU）的模型，尤其是方形权重层（如 q_proj, o_proj）占比较高的密集模型。修复后输出正确，精度经测试误差 <0.005。
- 风险标记：核心路径变更，数据契约变更

关联脉络

- PR #44113 [Bugfix] Fix MarlinFP8 weight transpose silently skipped for square matrices (N==K): 同一 bug 的替代修复方案，本 PR 采用更彻底的方法
- PR #38092 [PR unknown title]: 引入了 Marlin 核的布局检测和部分调用方的转置 workaround