PR #52966 完整报告

vllm-project/vllm

[Bugfix][Quantization] Support CT block FP8 with Marlin

合并时间：2026-08-20 03:41

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52966

执行摘要

- 一句话：Marlin 支持 CT block FP8 的 scale 命名兼容
- 推荐动作：值得精读。这是一个典型的数据契约兼容修复，展示了如何在共享内核抽象上兼容不同量化框架的 scale 命名差异。重点关注 _block_scale_name 的探测逻辑、process_weights_after_loading 与原属性名写回的配合，以及测试参数化如何覆盖 auto 与强制 Marlin 两条路径；同时可结合 #52182 与 #52908 理解前向修复与回退方案之间的取舍。

功能与动机

PR #52182 用公共 linear_backend / moe_backend 配置取代测试专用 VLLM_TEST_FORCE_FP8_MARLIN，使夜间 CI 显式选择 Marlin，暴露出 MarlinFP8ScaledMMLinearKernel 只读取 weight_scale_inv 的既有 gap。PR body 指出：原生 FP8 block 层暴露 weight_scale_inv，Compressed Tensors 层暴露 weight_scale；共享 block-kernel 抽象和 prepare_fp8_layer_for_marlin 都已支持两种约定，唯独 Marlin 内核无条件访问 weight_scale_inv，导致 nightly #84555 的 Qwen3-30B-A3B-Fp8-CT-Block-marlin 配置中两个 TP worker 在加载 QKV projection 时崩溃。本 PR 选择前向修复而非 #52908 的全量 revert，以保留 #52182 的配置清理成果。

实现拆解

1. 统一 block scale 命名契约：在 vllm/model_executor/kernels/linear/scaled_mm/marlin.py 中为 MarlinFP8ScaledMMLinearKernel 新增静态方法 _block_scale_name，通过 getattr 探测 layer 上的 weight_scale_inv 是否存在，决定返回 weight_scale_inv 还是 weight_scale。
2. 改造 process_weights_after_loading：block 量化分支先用 _block_scale_name 取得实际属性名，再把对应 scale 传给 process_fp8_weight_block_strategy，最后用 replace_parameter 以原属性名写回处理结果，从而保持属性命名不变。
3. 同步修改 apply_weights：block 量化分支改用 getattr(layer, self._block_scale_name(layer)) 读取 scale，保证推理读取与预处理写入使用同一命名探测逻辑。
4. 回归测试改造：tests/quantization/test_compressed_tensors.py 将 test_compressed_tensors_fp8_block_enabled 参数化为 linear_backend 的 ["auto", "marlin"]（非 CUDA 平台仅 auto），加载 RedHatAI/Qwen3-0.6B-FP8-BLOCK，并分别断

言 Marlin 路径（kernel 类型为 MarlinFP8ScaledMMLinearKernel、weight 为 int32、scale 保持 orig_dtype）与 auto 路径（Fp8BlockScaledMMLinearKernel、weight 为 fp8、scale 为 float32）的差异。

5. 配套验证与演进：提交历史包含一次实现提交和一次 merge main；Mergify 提示 pre-commit 失败后由提交者重跑并触发 Buildkite CI #84626 与 #84649；PR body 自述 B300 上两种 scale 命名 round trip 与强制 Marlin 模型生成均通过，但完整 H100 GSM8K MoE 集成清单尚未在该 head 上运行。

关键文件：

- vllm/model_executor/kernels/linear/scaled_mm/marlin.py（模块 量化内核；类别 source；类型 data-contract；符号 _block_scale_name, process_weights_after_loading, apply_weights）：修复核心：新增 _block_scale_name 统一 block scale 命名契约，修改 process_weights_after_loading 与 apply_weights，使 Marlin 内核同时支持 weight_scale_inv 与 weight_scale。
- tests/quantization/test_compressed_tensors.py（模块 回归测试；类别 test；类型 test-coverage；符号 test_compressed_tensors_fp8_block_enabled）：回归覆盖：把 test_compressed_tensors_fp8_block_enabled 参数化，同时验证 auto 与强制 Marlin 两条后端路径的 kernel 类型与 dtype 约束，直接防止该兼容 gap 复发。

关键符号：_block_scale_name, process_weights_after_loading, apply_weights, test_compressed_tensors_fp8_block_enabled

关键源码片段

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

修复核心：新增 _block_scale_name 统一 block scale 命名契约，修改 process_weights_after_loading 与 apply_weights，使 Marlin 内核同时支持 weight_scale_inv 与 weight_scale。

```
class MarlinFP8ScaledMMLinearKernel(FP8ScaledMMLinearKernel):
    """FP8 Marlin 内核，用于需要 Marlin 权重编码或缺少 FP8 硬件支持的场景。"""

    @staticmethod
    def _block_scale_name(layer: torch.nn.Module) -> str:
        # 两种 block scale 命名约定：
        # - 原生 FP8 block 层暴露 weight_scale_inv
        # - Compressed Tensors 层暴露 weight_scale
        # 这里探测实际存在的属性名，保证预处理写回和推理读取使用同一个属性。
        if getattr(layer, "weight_scale_inv", None) is not None:
            return "weight_scale_inv"
        return "weight_scale"

    def process_weights_after_loading(self, layer: torch.nn.Module) -> None:
        if self.block_quant:
            scale_name = self._block_scale_name(layer)
            weight, weight_scale = process_fp8_weight_block_strategy(
                layer.weight, getattr(layer, scale_name)
```

```
            )
            # 原地替换权重和 scale，并保留原有属性名，避免破坏后续读取约定。
            replace_parameter(layer, "weight", weight.data)
            replace_parameter(layer, scale_name, weight_scale.data)
        # 非 block 路径：调用方需传入 (K, N) 布局的权重。

        layer.input_scale = None
        prepare_fp8_layer_for_marlin(
            layer, self.size_k_first, input_dtype=self.marlin_input_dtype
        )
        del layer.input_scale

    def apply_weights(
        self,
        layer: torch.nn.Module,
        x: torch.Tensor,
        bias: torch.Tensor | None = None,
    ) -> torch.Tensor:
        if self.block_quant:
            # 与 process_weights_after_loading 使用同一命名探测逻辑。
            weight_scale = getattr(layer, self._block_scale_name(layer))
        else:
            weight_scale = layer.weight_scale
        return apply_fp8_marlin_linear(
            input=x,
            weight=layer.weight,
            weight_scale=weight_scale,
            workspace=layer.workspace,
            size_n=layer.output_size_per_partition,
            size_k=layer.input_size_per_partition,
            input_dtype=self.marlin_input_dtype,
            bias=bias,
        )
```

tests/quantization/test_compressed_tensors.py

回归覆盖：把 test_compressed_tensors_fp8_block_enabled 参数化，同时验证 auto 与强制 Marlin 两条后端路径的 kernel 类型与 dtype 约束，直接防止该兼容 gap 复发。

```
@pytest.mark.parametrize(
    "linear_backend", ["auto", "marlin"] if current_platform.is_cuda() else ["auto"]
)
def test_compressed_tensors_fp8_block_enabled(vllm_runner, linear_backend):
    model_path = "RedHatAI/Qwen3-0.6B-FP8-BLOCK"
    with vllm_runner(
        model_path, enforce_eager=True, linear_backend=linear_backend
    ) as llm:
        fp8_dtype = current_platform.fp8_dtype()

        def check_model(model):
```

```
        layer = model.model.layers[0]
        qkv_proj = layer.self_attn.qkv_proj
        assert isinstance(qkv_proj.quant_method, CompressedTensorsLinearMethod)
        assert isinstance(qkv_proj.scheme, CompressedTensorsW8A8Fp8)

        if linear_backend == "marlin":
            # 强制 Marlin 时，权重被重打包为 int32，block scale 保持原始 dtype。
            assert isinstance(
                qkv_proj.scheme.fp8_linear, MarlinFP8ScaledMMLinearKernel
            )
            assert qkv_proj.weight.dtype is torch.int32
            assert qkv_proj.weight_scale.dtype is qkv_proj.orig_dtype
        else:
            # auto 路径默认选原生 block kernel，权重保持 fp8、scale 为 float32。
            assert isinstance(
                qkv_proj.scheme.fp8_linear, Fp8BlockScaledMMLinearKernel
            )
            assert qkv_proj.weight.dtype is fp8_dtype
            assert qkv_proj.weight_scale.dtype is torch.float32
        assert len(qkv_proj.weight.shape) == 2
        assert len(qkv_proj.weight_scale.shape) == 2

        if linear_backend == "auto":
            input_quant_op = qkv_proj.scheme.fp8_linear.quant_fp8
            assert isinstance(input_quant_op, QuantFP8)
            assert input_quant_op._forward_method in (
                input_quant_op.forward_cuda,
                input_quant_op.forward_hip,
                input_quant_op.forward_xpu,
            )

    llm.apply_model(check_model)
    output = llm.generate_greedy("Hello my name is", max_tokens=4)
    assert output
```

评论区精华

该 PR 没有实质性的代码级 review 评论。claude[bot] 指出该 PR 来自 fork，自动审核被禁用，维护者可通过 @claude review 手动触发；issue 侧仅有 Mergify 的 pre-commit 失败提示与 mgoin 两次 /ci run 触发记录。PR body 中 mgoin 主动说明了与 #52908 的差异：本 PR 是前向修复，保留 #52182 并修复底层 CT/Marlin 不兼容，且让 H100 集成配置继续显式覆盖 Marlin；而 #52908 选择全量回退 #52182。AI 辅助说明部分也注明由提交者审阅并负责最终结果。

- 自动 review 状态 (other): 无代码级 review 讨论；后续 CI 与 pre-commit 状态由机器人与提交者管理。

风险与影响

- 风险：
  1. 数据契约判断依赖 getattr 探测：若某 CT 模型同时存在 weight_scale 与 weight_scale_inv 属性，_block_scale_name 会优先选择 weight_scale_inv，可能与预期不符；该逻辑仅在 block_quant 分支生效，影响面有限。
  2. Marlin 路径会把权重重打包为 int32 并替换 scale 值，现有回归测试只覆盖 Qwen3-0.6B 单模型，更大规模或更多 TP 配置的 CT block-FP8 模型仍需额外验证。
  3. 测试依赖远程模型 RedHatAI/Qwen3-0.6B-FP8-BLOCK，网络波动可能导致 CI 不稳定；且 marlin 参数分支只在 CUDA 平台启用，非 CUDA 平台覆盖不足。
  4. PR body 自述完整 H100 GSM8K 集成清单未在该 head 上运行，远程验证仍需确认。 - 影响：功能上修复了 Compressed Tensors block-FP8 模型显式选择 Marlin 时的加载崩溃，恢复 Qwen3-30B-A3B-Fp8-CT-Block-marlin 这类 H100 集成配置的可运行性。影响面集中在 MarlinFP8ScaledMMLinearKernel 的 block 处理路径，原生 Fp8BlockScaledMMLinearKernel 路径无行为变化。测试侧新增 auto / marlin 双路径参数化回归，为后端选择的分叉行为提供了防线。若本 PR 合并，#52908 的 full revert 方案应关闭，团队得以保留 #52182 的配置清理成果。 - 风险标记：数据契约兼容，核心内核路径变更，测试依赖远程模型，远程集成验证待确认

关联脉络

- PR #52182 Remove VLLM_TEST_FORCE_FP8_MARLIN to replace with linear_backend/moe_backend: 引入显式 Marlin 选择配置，暴露了 MarlinFP8ScaledMMLinearKernel 与 CT block FP8 之间的 scale 命名 gap；本 PR 是其前向修复。
- PR #52908 [Revert] Restore test-only FP8 Marlin selection: 竞争性方案：全量回退 #52182；PR body 明确说明本 PR 与它的差异并保留 #52182，若本 PR 合并则 #52908 应关闭。