

PR #54824 完整报告

vllm-project/vllm

[Bugfix] Restore `weight\_dtype` in `QuarkW8A8Fp8MoEMethod` to fix GPT-OSS FP8 MoE weight loading

合并时间: 2026-09-02 05:08

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/54824>

执行摘要

- 一句话: 恢复 QuarkW8A8Fp8MoEMethod 的 weight_dtype 属性以修复 FP8 MoE 权重加载崩溃。
- 推荐动作: 该 PR 是简单明确的 bugfix, 修复了重构引入的回归问题。建议快速审查并合并, 以保持 CI 绿灯和模型功能正常。值得精读以确认恢复的属性值 'fp8' 与上下游逻辑一致。

功能与动机

PR body 中明确说明: #52958 重构了 QuarkConfig/QuarkMoEMethod, 导致 QuarkW8A8Fp8MoEMethod.__init__ 中 self.weight_dtype 属性被意外移除。GptOssForCausalLM._load_weights_quark 依赖此属性通过 hasattr 检查来决定权重加载分支, 缺失属性导致返回 None, 进而跳过 FP8 分支, 引发 RuntimeError: The size of tensor a (2880) must match the size of tensor b (5760) at non-singleton dimension 0 错误。

实现拆解

- 问题定位: 识别到 QuarkW8A8Fp8MoEMethod 的 __init__ 方法在重构后缺少 weight_dtype 属性初始化, 导致下游 _get_moe_weight_dtype() 返回 None 而非 "fp8", 错误跳过 FP8 权重加载分支。
- 文件修改: 在文件 vllm/model_executor/layers/quantization/quark/quark_moe.py 的 QuarkW8A8Fp8MoEMethod.init__ 方法中, 于 super().__init__() 调用后添加 self.weight_dtype = "fp8" 单行代码。
- 修复原理: 恢复属性后, hasattr(quant_method, "weight_dtype") 检查为 True, _get_moe_weight_dtype() 正确返回 "fp8", 使得权重加载进入正确的 FP8 分支, 避免尺寸不匹配错误。
- 验证: 通过修改后的测试用例 cases/quantization/test_gpt_oss.py::test_gpt_oss_attention_quantization, CI 测试恢复通过。
- 配套改动: 无其他文件或测试变更。

关键文件:

- vllm/model_executor/layers/quantization/quark/quark_moe.py (模块 量化方法; 类别 source; 类型 data-contract; 符号 QuarkW8A8Fp8MoEMethod.init) : 恢复 QuarkW8A8Fp8MoEMethod 的 weight_dtype 属性, 修复 FP8 MoE 权重加载崩溃。

关键符号: QuarkW8A8Fp8MoEMethod.init

关键源码片段

[vllm/model_executor/layers/quantization/quark/quark_moe.py](#)

恢复 QuarkW8A8Fp8MoEMethod 的 weight_dtype 属性, 修复 FP8 MoE 权重加载崩溃。

```
def __init__(
    self,
    moe: FusedMoEConfig,
    weight_quant_key: QuantKey,
    activation_quant_key: QuantKey | None,
):
    super().__init__(moe, weight_quant_key, activation_quant_key)
    # 恢复 weight_dtype 属性, 确保下游 FP8 MoE 权重加载分支正确
    # 该属性被 PR#52958 重构意外移除, 导致 GPT-OSS 模型加载时
    # _get_moe_weight_dtype() 返回 None, 跳过 FP8 分支引发崩溃
    self.weight_dtype = "fp8"
    self.weight_qscheme = (
        "per_channel" if weight_quant_key == kFp8StaticChannelSym else "per_tensor"
    )
    self.input_qscheme = (
        "per_channel"
        if activation_quant_key == kFp8DynamicTokenSym
        else "per_tensor"
    )
    self.static_input_scales = activation_quant_key == kFp8StaticTensorSym
    per_tensor = (
        self.weight_qscheme == "per_tensor" and self.input_qscheme == "per_tensor"
    )
    per_channel = (
        self.weight_qscheme == "per_channel" and self.input_qscheme == "per_channel"
    )
    self.act_quant_group_shape = (
        GroupShape.PER_TOKEN if per_channel else GroupShape.PER_TENSOR
    )
    if not (per_tensor or per_channel):
        raise ValueError(
            "For FP8 Fused MoE layers, only per-tensor and per-channel "
            "scales for weights and activations are supported. Found "
            f"{self.weight_qscheme}, {self.input_qscheme}"
        ) # noqa E501

    self.fp8_backend, self.experts_cls = select_fp8_moe_backend(
        config=moe,
        weight_key=self.weight_quant_key,
        activation_key=self.activation_quant_key,
    )
```

评论区精华

无实质性技术讨论。自动审查 bot 提示这是 fork PR，自动审查已禁用。维护者 AndreasKaratzas 直接批准了 PR，未提供额外评论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更仅涉及添加单行属性设置，逻辑简单直接，影响路径明确。但需注意：
： 1) 其他量化方法类是否在类似重构中遗漏关键属性； 2) 依赖 `weight_dtype` 属性的下游代码是否覆盖所有场景。
- 影响：影响范围：仅影响 GPT-OSS 模型的 FP8 MoE 权重加载路径。影响程度：修复了关键崩溃错误，确保模型能正常加载和运行，CI 测试恢复通过，避免生产环境功能中断。
- 风险标记：属性缺失回归，单行修复

关联脉络

- PR #54573 [Fix] Fix FSE compatibility detection for Quark-produced models: 都涉及 Quark 量化模型的修复，可能影响 MoE 权重加载和量化配置检测路径。