PR #39799 完整报告

vllm-project/vllm

[ROCm][CI] Fix TestSiluMulGroupFp8QuantModel after W8A8 block linear refactor

合并时间：2026-04-25 10:21

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/39799

执行摘要

- 一句话：修复 ROCm 上 SiluMul+FP8 融合测试因重构而中断
- 推荐动作：值得精读，特别是了解在重构后如何联动调整测试代码的实践。关注点：平台区分（fnuz vs 非 fnuz）、猴子补丁技巧、操作列表与编译传递的对应关系。

功能与动机

W8A8 Block Linear 重构（PR#33892）将 W8A8BlockFp8LinearOp 替换为 TestFP8Layer，但 TestSiluMulGroupFp8QuantModel 的 forward 调用和操作列表未同步更新，导致 ROCm 平台上的测试失败。此外，在非 fnuz ROCm（如 MI355）上，融合模式期望 Triton 分组量化操作，但小测试形状下 use_triton 标志为 False，需强制启用。

实现拆解

1. 移除冗余权重 / 缩放张量：在 TestSiluMulGroupFp8QuantModel.__init__ 中删除了 self.w 和 self.wscale，因为 TestFP8Layer 在内部创建权重。
2. 强制 Triton 量化路径：在非 fnuz ROCm 平台上，通过猴子补丁（monkey-patch）强制 kernel.quant_fp8 的 use_triton=True，确保融合测试使用预期内核。
3. 更新操作列表：在 ops_in_model_before 中根据平台动态返回 rocm_aiter_ops.get_group_quant_op()（fnuz）或 torch.ops.vllm.triton_per_token_group_quant_fp8.default（非 fnuz），保证编译传递能够识别正确操作。
4. 调整容差阈值：为 TestSiluMulBlockQuantModel 在 ROCm 上设置更严格的容差（1e-3），同时 CUDA 上保持宽松（5e-2）以包容浮点计算差异。
5. 导入调整：将 rocm_aiter_ops 的导入从函数体内提升到文件顶部，避免重复导入。

关键文件：

- tests/compile/passes/test_silu_mul_quant_fusion.py（模块 编译测试；类别 test；类型 test-coverage；符号 TestSiluMulGroupFp8QuantModel, test_fusion_silu_and_mul_quant）：唯一修改的文件，修复了 SiluMul+FP8 融合测试的所有三个问题：forward 调用、操作列表、Triton 量化路径。

关键符号：未识别

关键源码片段

tests/compile/passes/test_silu_mul_quant_fusion.py

唯一修改的文件，修复了 SiluMul+FP8 融合测试的所有三个问题：forward 调用、操作列表、Triton 量化路径。

```
# tests/compile/passes/test_silu_mul_quant_fusion.py
# 在 TestSiluMulGroupFp8QuantModel.__init__ 中，猴子补丁强制 Triton 路径
if not current_platform.is_fp8_fnuz():
    kernel = self.w8a8_block_fp8_linear.kernel
    orig_quant = kernel.quant_fp8
    # 将所有 quant_fp8 调用强制使用 use_triton=True
    kernel.quant_fp8 = lambda *a, use_triton=False, **kw: orig_quant(
        *a, use_triton=True, **kw
    )

# ops_in_model_before 根据平台动态返回量化操作
# 对于 fnuz ROCm 使用 AITER 操作，否则使用 Triton 操作
# 这是因为融合模式需要精确匹配预期操作列表
def ops_in_model_before(self):
    return [
        SILU_MUL_OP if self.enable_silu_mul_custom_op else torch.ops.aten.mul,
        rocm_aiter_ops.get_group_quant_op()
        if current_platform.is_fp8_fnuz()
        else torch.ops.vllm.triton_per_token_group_quant_fp8.default,
    ]

# 在 test_fusion_silu_and_mul_quant 中区分模型类型调整容差
# ROCm 上 BlockQuant 模型使用更严格的 1e-3，CUDA 上保持 5e-2
elif isinstance(model, TestSiluMulBlockQuantModel):
    if current_platform.is_rocm():
        atol, rtol = 1e-3, 1e-3
    else:
        atol, rtol = 5e-2, 5e-2
```

评论区精华

该 PR 没有实质性的人工 review 讨论。自动机器人 gemini-code-assist[bot] 提供了摘要性评论，但无反馈意见。审核者 tjtanaa 直接批准（LGTM）。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅限于测试文件 tests/compile/passes/test_silu_mul_quant_fusion.py，不涉及生产代码。主要风险在于：
  - 如果其他测试模式或实际生产路径也依赖于类似的 TestFP8Layer 调用模式，可能因未同步更新而失败。但本 PR 已针对特定模型修复。
  - 强制 use_triton=True 可能在小 shape 下引入 Triton 代码路径，但测试范围有限，不易产生副作用。

- 影响：正面影响：恢复了 ROCm 平台上 SiluMul 与 FP8 融合测试的正确性，确保编译器融合传递在 AMD GPU 上按预期工作。 影响范围：仅影响测试文件，用户不受直接影响。对开发团队，维护了 CI 的稳定性。
- 风险标记：仅测试文件变更

关联脉络

- PR #33892 W8A8 block linear refactor: 本 PR 修复的回归问题正是由该重构引入，是直接关联的上游 PR。