

PR #46410 完整报告

vllm-project/vllm

[ROCm][CI] fix fp8 range in vit_fp8_quant

合并时间: 2026-06-24 00:34

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/46410>

执行摘要

- 一句话: 修复 AMD FNUZ 平台 FP8 量化测试范围错误
- 推荐动作: 建议合并。这是对 CI 测试的正确性修复, 改动小且经过 reviewer 审核, 应快速合入以恢复 ROCm CI 稳定性。

功能与动机

PR body 指出 "Fix fp8 min/max range for FNUZ platform", 并明确修复测试 `kernels/core/test_vit_fp8_quant.py::test_quantize_contiguous`。该测试在 AMD ROCm 平台上失败, 因为 `torch.finfo(fp8_dtype).max` 返回的是 OCP FP8 的最大值 240, 而 AMD MI300 等 FNUZ 平台实际钳位到 224, 导致参考实现与 kernel 行为不一致, 测试断言失败。

实现拆解

1. 修改导入: 在 `tests/kernels/core/test_vit_fp8_quant.py` 中增加从 `vllm.model_executor.layers.quantization.utils.quant_utils` 导入 `get_fp8_min_max` 函数。
2. 替换范围获取方式: 将 `_naive_fp8_quantize` 函数中原来使用 `torch.finfo(fp8_dtype).max` 和 `-max` 获取范围的方式, 改为调用 `get_fp8_min_max()` 返回平台相关的 `(fp8_min, fp8_max)`。
3. 未修改 kernel 或量化逻辑: 该 PR 仅修正了参考实现的饱和范围, kernel 本身已使用 `get_fp8_min_max`, 因此统一后测试能够正确验证。

关键文件:

- `tests/kernels/core/test_vit_fp8_quant.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 `_naive_fp8_quantize`): 唯一变更文件, 修复了 FP8 参考量化函数的饱和范围, 使其与平台对齐, 确保测试在 AMD FNUZ 上通过。

关键符号: `_naive_fp8_quantize`

关键源码片段

`tests/kernels/core/test_vit_fp8_quant.py`

唯一变更文件, 修复了 FP8 参考量化函数的饱和范围, 使其与平台对齐, 确保测试在 AMD FNUZ 上通过。

```
# tests/kernels/core/test_vit_fp8_quant.py
```

修改后：使用 `get_fp8_min_max()` 获取平台相关的 FP8 饱和范围

```
def _naive_fp8_quantize(
    tensor: torch.Tensor, scale: torch.Tensor, skip_scale: bool
) -> torch.Tensor:
    """Reference FP8 quantization in PyTorch."""
    fp8_dtype = current_platform.fp8_dtype()
    # 原来用 torch.finfo(fp8_dtype).max 得到 OCP 标准最大值 240,
    # 但 AMD FNUZ 平台 clamp 至 +/-224, 导致参考与 kernel 不一致。
    # 现在改用 get_fp8_min_max(), 该函数会根据平台返回正确的范围。
    fp8_min, fp8_max = get_fp8_min_max()

    x = tensor.float()
    if not skip_scale:
        x = x / scale.item()
    x = x.clamp(fp8_min, fp8_max)
    return x.to(fp8_dtype)
```

评论区精华

Review 中 yewentao256 首先建议扩展 `_naive_fp8_quantize` 的 docstring 说明原因，随后又撤回该建议 ("We don't need the comment here")。最终提交采纳了不添加多余注释的版本。整个过程简洁，无重大争议。

- 是否需要在 docstring 中说明范围变更原因 (style): 维持原 docstring 不变，不添加额外注释。

风险与影响

- 风险：风险极低。变更仅涉及测试文件，修改了两个赋值语句和一个 import，不影响生产代码。唯一风险是如果未来 `get_fp8_min_max` 的行为发生变化，测试的参考实现也会随之变化，但这正是测试与 kernel 保持一致所需要的。
- 影响：直接影响：使得 `test_vit_fp8_quant.py` 在 AMD ROCm/CPU FNUZ 平台上正确通过，消除 CI 中的假阴性失败。间接影响：无，因为未改动任何生产逻辑。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR