PR #52647 完整报告

vllm-project/vllm

[ROCm][CI] Expand AITER W4A4 MoE Coverage

合并时间：2026-08-18 04:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52647

执行摘要

- 一句话：扩展 AITER W4A4 MoE 测试，补回退用例与激活量化参考
- 推荐动作：值得快速浏览。虽然没有生产代码，但两层设计有借鉴价值：一是用“环境变量 + 模块级标志双控制”的 fixture 在同一台真机上同时验证开启与回退两条路径；二是 oracle 参考路径如何精确模拟激活量化误差。对负责 ROCm 内核后端选择的工程师有直接参考价值。

功能与动机

PR body 明确两个目标：其一，在 GFX950 上用显式 AITER 开 / 关 fixture 测试 w4a4 后端选择（“Test gfx950 w4a4 backend selection with explicit AITER on/off fixtures, including a no-AITER emulation fallback test”），覆盖无 AITER 时的 EMULATION 回退；其二，扩展现有 mxfp4 MoE oracle 测试以覆盖 AITER_MXFP4_MXFP4，并在参考路径中加入 mxfp4 激活量化（“with mxfp4 activation quantization in the reference path”）。背景是 AITER 新增了激活也走 MXFP4 的 W4A4 MoE 后端，而既有测试只覆盖 BF16/FP8 激活变体，且只能被动依赖运行环境中 AITER 是否启用，无法验证“禁用 AITER 时正确回退”这一关键路径。

实现拆解

1. GFX950 后端选择测试的可控化（tests/quantization/test_gfx950_moe.py）

1. 模块级探测从 rocm_aiter_ops.is_fused_moe_enabled() 改为 is_aiter_found_and_supported()（从 vllm._aiter_ops 导入），把“AITER 库可用”与“进程内已启用”两个概念解耦，为后续通过 fixture 动态切换 AITER 开关铺路。
2. 新增 set_rocm_aiter(monkeypatch, enabled)：同时写 VLLM_ROCM_USE_AITER 与 VLLM_ROCM_USE_AITER_MOE 环境变量，并 monkeypatch rocm_aiter_ops._AITER_ENABLED 与 _FMOE_ENABLED 两个模块级标志，保证 select_mxfp4_moe_backend 运行时读到测试期望的状态。
3. test_w4a4_dispatches_to_aiter 改为依赖 enable_rocm_aiter fixture，跳过条件从“AITER 已启用”放宽为“AITER 受支持”；新增 test_w4a4_falls_back_without_aiter，在 disable_rocm_aiter 下断言 W4A4 回退到 Mxfp4MoeBackend.EMULATION，从而在真机上同时覆盖“走 AITER 内核”与“回退实现”两条分支。

2. oracle 参考路径加入 mxfp4 激活量化（tests/kernels/moe/test_ocp_mx_moe.py）

1. 新增 mxfp4_quant_dequant：先用 dynamic_mxfp4_quant 按 block 做动态量化，再用 upcast_from_mxfp 反量化回 bf16/fp32，精确模拟内核中激活量化的精度损失。
2. reference_moe 支持 act_type == "mxfp4"：在 MLP #1 输入前与激活后（MLP #2 前）各做一次量化 - 反量化；原先只有 mxfp8 分支在激活后做量化。
3. ROCM_BACKEND_CONFIGS 新增 AITER_MXFP4_MXFP4（activation=SILU、act_type=mxfp4、rtol=1.0、percent=0.8），并把参考路径的 act_type 改为从配置读取（默认 bf16），使精度断言随后端变体变化。

3. 测试与 CI 配套

无 schema、配置或部署改动；CI 由维护者 /ci run 触发 Buildkite CI #84252；PR body 声明两个测试文件在 MI355 本地通过。

关键文件：

- tests/quantization/test_gfx950_moe.py（模块 后端选择；类别 test；类型 test-coverage；符号 set_rocm_aiter, enable_rocm_aiter, disable_rocm_aiter, test_w4a4_dispatches_to_aiter）：本次 PR 的核心：将 AITER 探测与启用解耦，引入 enable/disable fixture 显式控制 AITER 状态，并新增无 AITER 时回退 EMULATION 的用例，覆盖后端选择的两条关键分支。
- tests/kernels/moe/test_ocp_mx_moe.py（模块 精度测试；类别 test；类型 test-coverage；符号 mxfp4_quant_dequant, reference_moe, test_rocm_mxfp4_moe_oracle）：扩展 oracle 参考实现支持 mxfp4 激活量化，并新增 AITER_MXFP4_MXFP4 后端配置，使 W4A4 内核精度可被真实数值校验。

关键符号：set_rocm_aiter, enable_rocm_aiter, disable_rocm_aiter, test_w4a4_dispatches_to_aiter, test_w4a4_falls_back_without_aiter, mxfp4_quant_dequant, reference_moe, test_rocm_mxfp4_moe_oracle

关键源码片段

tests/quantization/test_gfx950_moe.py

本次 PR 的核心：将 AITER 探测与启用解耦，引入 enable/disable fixture 显式控制 AITER 状态，并新增无 AITER 时回退 EMULATION 的用例，覆盖后端选择的两条关键分支。

```
# 通过 fixture 显式控制 AITER 开关，让同一台 GFX950 上能同时覆盖
# “走 AITER 内核”与“回退 EMULATION”两条分支。
def set_rocm_aiter(monkeypatch: pytest.MonkeyPatch, enabled: bool) -> None:
    value = "1" if enabled else "0"
    # 同时改环境变量与模块级标志位：
    # 环境变量是 select_mxfp4_moe_backend 的运行时判定依据，
    # 模块级 _AITER_ENABLED / _FMOE_ENABLED 用于覆盖进程内缓存状态。
    monkeypatch.setenv("VLLM_ROCM_USE_AITER", value)
    monkeypatch.setenv("VLLM_ROCM_USE_AITER_MOE", value)
    monkeypatch.setattr(rocm_aiter_ops, "_AITER_ENABLED", enabled)
```

```
    monkeypatch.setattr(rocm_aiter_ops, "_FMOE_ENABLED", enabled)


@pytest.fixture
def enable_rocm_aiter(monkeypatch: pytest.MonkeyPatch):
    set_rocm_aiter(monkeypatch, True)


@pytest.fixture
def disable_rocm_aiter(monkeypatch: pytest.MonkeyPatch):
    set_rocm_aiter(monkeypatch, False)


@pytest.mark.skipif(not ROCM_GFX950, reason="Requires GFX950 (mi355x)")
# 只要求 GFX950、不要求 AITER 存在——正好验证"无 AITER 环境"的兜底路径。
def test_w4a4_falls_back_without_aiter(
    mxfp4_oracle_config,
    disable_rocm_aiter,
):
    config = _make_w4a4_moe_config()
    backend, experts_cls = select_mxfp4_moe_backend(
        config, activation_key=kMxfp4Dynamic
    )
    # 关闭 AITER 后，W4A4 必须回退到 EMULATION 实现而不是报错。
    assert backend == Mxfp4MoeBackend.EMULATION
    assert experts_cls is not None
```

tests/kernels/moe/test_ocp_mx_moe.py

扩展 oracle 参考实现支持 mxfp4 激活量化，并新增 AITER_MXFP4_MXFP4 后端配置，使 W4A4 内核精度可被真实数值校验。

```
def mxfp4_quant_dequant(x: torch.Tensor) -> torch.Tensor:
    """参考路径里的 mxfp4 激活量化。

    先用 dynamic_mxfp4_quant 按 block 做动态量化，再用 upcast_from_mxfp
    反量化回 bf16，模拟 AITER_MXFP4_MXFP4 内核中"激活也走 MXFP4"
    带来的精度损失，让参考实现与真实内核对齐。
    """
    shape = x.shape
    # flatten(0, -2) 保证最后一维连续，便于 scale block 对齐。
    quantized, scale = dynamic_mxfp4_quant(x.to(torch.bfloat16).flatten(0, -2))
    return (
        upcast_from_mxfp(quantized.view(torch.uint8), scale, torch.bfloat16, axis=-1)
        .reshape(shape)
        .float()
    )


# reference_moe 中新增的 mxfp4 分支（节选）：
```

```
# MLP #1 之前，输入激活先做一次量化 - 反量化（原先只有权重是 MXFP4）。
t = hidden_states.clone()
if act_type == "mxfp4":
    t = mxfp4_quant_dequant(t)

# 激活之后、MLP #2 之前再补一次；原先只有 mxfp8 分支做激活后量化。
if act_type == "mxfp8":
    t_quantized, t_scale = mxfp8_quantize(
        t.to(torch.bfloat16), is_sf_swizzled_layout=False
    )
    t = mxfp8_dequantize(t_quantized, t_scale)
elif act_type == "mxfp4":
    t = mxfp4_quant_dequant(t)
```

评论区精华

review 讨论非常精简：claude[bot] 提示该 PR 来自 fork，自动化 review 被禁用，维护者可用 `@claude review` 手动触发（最终未执行）；维护者 AndreasKaratzas 直接 APPROVED 并仅给出 "LGTM"，没有任何质疑性 comment。需要留意的是，评论中未讨论两处潜在问题：一是 `_AITER_ENABLED` / `_FMOE_ENABLED` 这类私有标志 monkeypatch 的脆弱性；二是 `AITER_MXFP4_MXFP4` 的 `rtol=1.0` 相对宽松，是否足够证明内核正确性。

- fork PR 自动化 review 未启用 (other): 未触发额外 review，最终由维护者 AndreasKaratzas 人工 APPROVED。
- CI 触发与本地验证 (question): CI 已触发，review 无进一步疑问后合并。

风险与影响

- 风险：生产风险为零，全部改动位于测试文件。主要风险点：1）set_rocm_aiter 依赖 rocm_aiter_ops._AITER_ENABLED / _FMOE_ENABLED 私有属性，若 AITER 封装改名或改为惰性读取环境变量，测试可能静默跳过或误判，属于对内部实现的耦合；2）AITER_MXFP4_MXFP4 的 rtol=1.0 允许 100% 相对误差且 percent=0.8，精度断言偏弱，存在“通过但精度损失明显”的假阴性风险；3）两条新用例的 skip 条件依赖 GFX950 与 AITER 支持，普通 CI 机器不会执行，实际守护效果取决于 ROCm 硬件 CI 是否常驻 MI355；4）is_aiter_found_and_supported() 是新引入的探测 API，若其语义变化会影响测试有效性。
- 影响：影响面局限于 ROCm/GFX950 专属测试矩阵：为 AITER_MXFP4_MXFP4 后端选择与精度提供回归基线，对运行用户与服务端行为零影响。对团队而言，补齐了 W4A4 激活量化路径的验证缺口，后续在 MI355 上的 AITER 内核改动会立即被这两条新用例守护；同时它也是近两周 ROCm CI 稳定性加固工作的一部分。
- 风险标记：仅测试改动，依赖私有标志位，精度阈值偏宽，硬件专属测试

关联脉络

- PR #52566 [ROCm][CI] Restore Torch defaults and type DSV4 scratch buffers: 同属 ROCm CI 测试稳定性修复方向，本 PR 合入后共用 MI355 硬件 CI 资源，体现 ROCm 测

试矩阵的持续加固。

- PR #52502 [Hardware][NVIDIA] Add GB10 fused-MoE fp8 tuning configs (E=256, E=512): 同属 MoE 量化后端配置与精度验证脉络，说明 fused-MoE 后端矩阵正在多平台扩展，本 PR 的 AITER_MXFP4_MXFP4 用例是 ROCm 侧的补位。
- PR #51114 [Perf][MoE] Optimize deepep_v2 receiver CPU Overhead: 同属 MoE 内核实现 / 验证链路，体现 fused MoE 多后端并存下测试配套的演进方向。