

PR #49375 完整报告

vllm-project/vllm

[ROCm][CI] Add More AITER quantization/MoE kernel tests

合并时间: 2026-08-06 05:27

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49375>

执行摘要

- 一句话: 新增 ROCm AITER 量化与 MoE 内核测试
- 推荐动作: 值得精读, 尤其是三个新测试文件中的统计式精度断言设计 (`_assert_close_budget`、`_assert_group_quant_quality`、`_assert_abs_error_budget`、`_assert_rel_error_budget`) 和 AITER enablement 门控测试 (`test_rocm_aiter_fp4_enablement_follows_env_and_arch`、`test_aiter_fp8bmm_enabled_api_respects_env`), 对理解 ROCm AITER 的环境变量契约非常有帮助, 也可作为其他平台内核测试的模板。环境恢复 fixture `_restore_rocm_env_state` 是测试隔离的良好实践。

功能与动机

PR body 明确指出: 此前 ROCm 专属的 FP8、MXFP4/FP4 和 AITER fused MoE 代码路径在 AMD 硬件上几乎没有内核级测试。AMD 维护者 AndreasKaratzas 在评论中回应审查者 yewentao256 时说明: "There is no coverage right now for some critical AITER kernels and so we are adding that similar to other kernel test group that upstream has. These by the way are ROCm only test groups so they should not be a burden for upstream CI." 即为了补齐关键 AITER 内核的测试覆盖, 且这些测试组仅面向 ROCm, 不会给上游 CI 增加负担。

实现拆解

1. 新增 `tests/kernels/moe/test_rocm_aiter_moe.py` (+1079 行): 覆盖 ROCm AITER fused-MoE 自定义算子路径, 包括 custom-op 注册与 fake-tensor 支持、VLLM_ROCM_USE_AITER_MOE 与 shared experts 的 enablement 门控、BF16 精度、gfx950 专属 MXFP4 W4A16 MoE 精度与确定性、FP8 group-quant 激活质量, 以及 gfx942/gfx950 代表性 shape 覆盖。测试通过 Float32 mask-based MoE 参考实现 `ref_moe_forward` 与内核输出对比, 并用 `_assert_close_budget/_assert_group_quant_quality` 做统计式精度断言 (pass_rate、max_violation、mean 误差上限)。
2. 新增 `tests/kernels/quantization/test_rocm_mxfp4.py` (+638 行): 覆盖 ROCm AITER FP4/MXFP4 路径, 包括环境变量默认值与 enablement 门控 (VLLM_ROCM_USE_AITER_FP4_ASM_GEMM、VLLM_ROCM_USE_AITER_FP4BMM, 且依赖 gfx950 硬件检查)、vLLM 公开 MXFP4 QDQ helper 在大 shape 下与 torch 参考精确一致、Triton dynamic_mxfp4_quant 输出格式 (packed uint8 FP4 与 E8M0 scale

shape) 与确定性、以及 FP4 GEMM、preshuffled-scale、hardware-quant、skinnydecode 路径。

3. 新增 tests/kernels/quantization/test_rocm_fp8.py (+541 行) : 覆盖 ROCm FP8 量化与内核, 包括三个环境变量默认值 / 覆盖测试 (VLLM_ROCM_FP8_PADDING、VLLM_ROCM_USE_AITER_FP8BMM、VLLM_ROCM_FP8_MFMA_PAGE_ATTN) 、_maybe_pad_fp8_weight 是否尊重环境开关、AITER FP8BMM enablement API、torch._scaled_mm 与反量化参考的误差预算对比, 以及 FP8 paged attention 与朴素参考实现 _ref_paged_attention 的对比 (含 GQA repeat_interleave 与 causal mask) 。
4. 修改 tests/kernels/utils.py (+57 行) : 新增共享工具 _assert_accurate (pass_rate + max_violation_factor + mean 误差三重断言)、_assert_deterministic (多次运行 bitwise 相等校验, 可处理返回 Tensor 或 Tensor 元组 / 列表的情况) 。
5. 修改 .buildkite/test-amd.yaml (+20/-2) : 为 MI300 与 MI355 平台的 Kernels MoE Test 和 Kernels Quantization Test 更新 source_file_dependencies, 加入 csrc/rocm/、tests/kernels/utils.py、tests/kernels/quant_utils.py、vllm/_custom_ops.py、vllm/envs.py、vllm/v1/attention/backends/rocm_aiter_fa.py 等文件, 并移除对 test_rocm_skinny_gemms.py 的显式依赖 (该文件恢复保留, 但不再作为依赖锚点) 。

关键文件:

- tests/kernels/moe/test_rocm_aiter_moe.py (模块 内核测试; 类别 test; 类型 test-coverage; 符号 _reload_envs, _restore_rocm_env_state, _assert_aiter_supported, _format_observed_rate) : 本 PR 最大新增文件 (+1079 行), 覆盖 ROCm AITER fused-MoE 完整路径: enablement 门控、BF16/MXFP4 精度、FP8 group-quant 质量、确定性, 是本次测试补强的核心。
- tests/kernels/quantization/test_rocm_mxfp4.py (模块 量化测试; 类别 test; 类型 test-coverage; 符号 _reload_envs, _restore_rocm_env_state, _assert_aiter_supported, _format_observed_rate) : 新增 638 行, 覆盖 gfx950 关键 FP4/MXFP4 路径: AITER enablement 门控 (依赖 on_gfx950 硬件判断)、MXFP4 QDQ 参考一致性、Triton quant 输出格式与确定性。
- tests/kernels/quantization/test_rocm_fp8.py (模块 量化测试; 类别 test; 类型 test-coverage; 符号 _reload_envs, _restore_rocm_env_state, _quantile, _format_observed_rate) : 新增 541 行 FP8 覆盖: 环境开关契约 (含 _maybe_pad_fp8_weight 行为差异)、scaled_mm 反量化参考对比、FP8 paged attention 与手写参考实现对比。
- tests/kernels/utils.py (模块 测试工具; 类别 test; 类型 test-coverage; 符号 _assert_accurate, _assert_deterministic, _collect) : 新增被三个测试文件共享的断言工具 _assert_accurate 与 _assert_deterministic, 是本次测试基础设施的关键通用化。
- .buildkite/test-amd.yaml (模块 CI 配置; 类别 config; 类型 configuration) : 更新 MI300/MI355 的 Kernels MoE 与 Quantization 测试组依赖清单, 确保新增测试文件与相关源码变更能触发对应 CI 分组。

关键符号: _assert_accurate, _assert_deterministic, _assert_close_budget, _assert_group_quant_quality, _assert_abs_error_budget, _assert_rel_error_budget, _ref_paged_attention, ref_moe_forward, test_rocm_aiter_fp4_enablement_follows_env_a

```
nd_arch, test_aiter_fp8bmm_enabled_api_respects_env,
_restore_rocm_env_state, _make_aiter_mxfp4_moe_case,
test_aiter_dynamic_mxfp4_quant_output_format
```

关键源码片段

tests/kernels/utils.py

新增被三个测试文件共享的断言工具 `_assert_accurate` 与 `_assert_deterministic`，是本次测试基础设施的关键通用化。

```
def _assert_accurate(
    actual: torch.Tensor,
    expected: torch.Tensor,
    atol: float,
    rtol: float = 0.0,
    pass_rate: float = 0.99999,
    max_violation_factor: float = 3.0,
) -> None:
    """统计式精度断言: 通过率 + 最大误差 + 均值误差三重约束。

    内核数值路径允许少数离群点，因此不要求所有元素都落在 atol 内；
    但离群幅度不得超过 max_violation_factor * atol，且均值误差
    需小于 atol * 0.25，防止整体系统性偏移。
    """
    a = actual.detach().float().flatten()
    e = expected.detach().float().flatten()

    abs_err = (a - e).abs()
    tol = atol + rtol * e.abs()

    rate = (abs_err <= tol).float().mean().item()
    assert rate >= pass_rate, (
        f"Accuracy pass rate {rate:.6f} < {pass_rate} "
        f"(atol={atol}, rtol={rtol})"
    )

    max_err = abs_err.max().item()
    assert max_err <= max_violation_factor * atol, (
        f"Max absolute error {max_err:.6f} exceeds "
        f"{max_violation_factor} * atol={atol}"
    )

    mean_err = abs_err.mean().item()
    assert mean_err <= atol * 0.25, (
        f"Mean absolute error {mean_err:.6f} >= atol * 0.25 = {atol * 0.25:.6f}"
    )

def _assert_deterministic(
    fn,
```

```

*args,
n_runs: int = 4,
**kwargs,
) -> None:
    """验证多次调用产生 bitwise 一致的张量输出。"""

def _collect(result: Any) -> list[torch.Tensor]:
    # 支持返回单个 Tensor 或 Tensor 元组 / 列表，统一收集为列表
    if isinstance(result, torch.Tensor):
        return [result.detach().clone()]
    if isinstance(result, (tuple, list)):
        return [
            t.detach().clone()
            for t in result
            if isinstance(t, torch.Tensor)
        ]
    raise TypeError(f"Unexpected return type {type(result)}")

reference = _collect(fn(*args, **kwargs))

for run in range(1, n_runs):
    outputs = _collect(fn(*args, **kwargs))
    for idx, (ref, out) in enumerate(zip(reference, outputs)):
        assert torch.equal(ref, out), (
            f"Run {run}: output[{idx}] differs from run 0 "
            f"(max diff = "
            f"{(out.float() - ref.float()).abs().max().item():.2e})"
        )

```

评论区精华

审查者 yewentao256 最初给出 DISMISSED 评审，质疑："Is there any specific bug related to these CI tests? If not we prefer not adding unit tests now as our CI has been quite heavy." 担心上游 CI 负担过重。AMD 维护者 AndreasKaratzas 回复解释这些是 ROCm 专属测试组，且目前关键 AITER 内核零覆盖，并最终 APPROVE: "LGTM let's begin with this foundation and continue building upon these new tests". claude[bot] 因 fork PR 自动审查被禁用，未产生实质讨论。

- 是否应新增这些 ROCm CI 测试 (question): AMD 维护者 AndreasKaratzas 解释这些是 ROCm 专属测试组，不上游负担；关键 AITER 内核当前零覆盖，需补齐。最终 yewentao256 的评审被 DISMISSED, AndreasKaratzas 批准合并。

风险与影响

- 风险:

1. CI 时长风险：新增三个测试文件共约 2200 行且含大量 shape 参数化用例，虽然仅注册到 AMD 硬件分组，但 MI300/MI355 CI 任务为 optional 标记，超时时间 180 分钟，测试规模扩大可能导致 AMD CI 变慢或超时。

2. 环境状态泄漏风险：测试通过 `importlib.reload(vllm.envs)` 和 `rocm_aiter_ops.refresh_env_variables()` 修改全局单例状态，虽然已用 `autouse` fixture 在每个测试后恢复，但如果 AITER 内部缓存了其他 env 相关状态（如 `is_aiter_found_and_supported` 的结果），可能造成测试间相互污染。
3. 精度断言参数风险：测试中大量使用统计式断言（如 `pass_rate 0.9999`、`mean` 误差限 `atol*0.25`、`max_violation_factor 3.0`），这些参数是针对特定硬件和 AITER 版本调出来的。未来 AITER 内核更新或新硬件（如后续 gfx 系列）上可能出现非内核 bug 导致的随机 CI 失败，需要维护者持续校准阈值。
4. 参考实现正确性风险：FP8 paged attention 的参考实现 `_ref_paged_attention` 和 MoE 的 `ref_moe_forward` 为测试内手写实现，若参考实现本身有 bug（如 `mask diagonal` 值、GQA 重复逻辑），则测试会把内核错误和参考错误混淆。
 - 影响：影响范围：仅涉及测试文件和 AMD CI 配置，不修改任何 vLLM 运行时源码，对用户无直接功能影响。对团队的直接影响是 AMD (MI300/MI355) CI 的 `Kernels` MoE/Quantization 测试组依赖关系更完整，后续这些测试会真实运行在 AMD 硬件上。价值在于为 ROCm AITER 量化与 MoE 路径建立了可持续的回归防线，尤其为 gfx950 的新 FP4 硬件路径提供了定向验证，未来内核改动可通过这些测试快速发现数值回退。
 - 风险标记：纯测试变更，CI 时长可能增加，精度阈值需持续校准，环境变量全局状态依赖

关联脉络

- PR #51173 [ROCm][CI] Keep rocprofiler-sdk out of DeepEP HT MoE test workers: 同为 ROCm CI 测试稳定性修复，且都涉及 `.buildkite/test-amd.yaml` 中 `Kernels` MoE/FP8 MoE 测试组的配置。
- PR #51083 [ROCm] Relax MLA rope+cache test tolerances for bf16: 同为 ROCm 内核测试容差调整，反映 ROCm 测试维护中精度阈值校准的常见主题。
- PR #51176 Revert [Misc] Avoid importing `nixl_ep` on every vllm serve config (#50879): 同为 ROCm 相关回归修复，涉及 AITER 相关导入与 CI 失败问题，与本 PR 的 `rocm_aiter` 测试主题相关。