

PR #54984 完整报告

vllm-project/vllm

[CI/Build][ROCm] Guard the two CUDA-only tests in test_bf16_skinny_gemm

合并时间: 2026-09-03 02:21

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/54984>

执行摘要

- 一句话: 修复 BF16 skinny GEMM 测试中两个隐含 CUDA 假设的测试用例在 ROCm 上失败的问题。
- 推荐动作: 这个 PR 值得精读, 尤其是其 PR body 中对失败原因的详细剖析和修复方案的论证。虽然变更规模小, 但它是一个关于如何正确编写跨平台测试 (特别是处理可选依赖和平台特定代码路径) 的优秀范例。核心决策点在于使用 `importorskip` 与 `skipif` 的区别, 以及理解平台分发如何影响测试前提。

功能与动机

修复 ROCm CI 中 (MI355) `Miscellaneous Kernels` 作业的失败。失败由 PR #50519 扩展 ROCm CI 覆盖范围后暴露出来, 该作业运行了 `test_bf16_skinny_gemm.py`。PR body 明确指出, “Both failures indicate a bug in the code under test. Both are unstated CUDA assumptions inside the tests themselves.” (两次失败都未指明被测代码存在 bug, 而是测试用例自身未声明的 CUDA 假设导致的)。

实现拆解

1. 导入与平台检测: 在测试文件顶部新增 `from vllm.platforms import current_platform`, 以便在测试函数中判断当前平台。
2. 修复 `test_kda_qkvg_autotune_enables_full_overlap`: 将函数体内的 `import flashinfer.gemm` 裸导入替换为 `flashinfer.gemm = pytest.importorskip("flashinfer.gemm")`。这确保了当 `flashinfer` 未安装时 (始终在 ROCm 上, 有时在 CUDA 上), 测试会被标记为跳过并显示原因, 而不是抛出 `ImportError`。
3. 修复 `test_fallback_preserves_default_method`: 在函数定义前添加 `@pytest.mark.skipif` 装饰器, 条件为 `not current_platform.is_cuda()`。其跳过原因是: 基准 `UnquantizedLinearMethod` 是按平台分发的, 在 ROCm 上会调用 `vllm::rocm_unquantized_gemm` 自定义算子, 该算子没有 CPU 内核, 因此测试中使用的 CPU 张量回退无法在该路径下执行。
4. 测试验证: 在 MI355X (gfx950) 上运行测试套件, 确认两个测试用例被正确跳过, 且其余测试通过, 无失败。

关键文件:

- tests/kernels/test_bf16_skinny_gemm.py (模块 内核测试; 类别 test; 类型 test-coverage; 符号 test_kda_qkvg_autotune_enables_full_overlap, test_fallback_preserves_default_method) : 这是本次变更的唯一文件, 包含了两个需要修复的测试用例及其修复代码。

关键符号: test_kda_qkvg_autotune_enables_full_overlap,
test_fallback_preserves_default_method

关键源码片段

tests/kernels/test_bf16_skinny_gemm.py

这是本次变更的唯一文件, 包含了两个需要修复的测试用例及其修复代码。

```
# 修复 1: 使用 pytest.importorskip 处理可选依赖 flashinfer
# 当 flashinfer 未安装时 (如在 ROCm 环境), 测试将被标记为跳过并显示原因。
def test_kda_qkvg_autotune_enables_full_overlap(
    monkeypatch: pytest.MonkeyPatch,
) -> None:
    # 替换原来的裸 `import flashinfer.gemm`
    flashinfer_gemm = pytest.importorskip("flashinfer.gemm")

    from vllm.models.kimi_k3.nvidia.kda import KimiK3DeltaAttention

    # ... 函数其余部分使用 flashinfer_gemm 变量 ...
    monkeypatch.setattr(flashinfer_gemm, "mm_bf16", fake_mm_bf16)

# 修复 2: 使用 @pytest.mark.skipif 门控整个测试函数
# 该测试验证的是平台分发回退逻辑, 但其前提 (使用 CPU 张量触发回退) 在 ROCm 上不成立。
@pytest.mark.skipif(
    not current_platform.is_cuda(),
    reason="The base UnquantizedLinearMethod is platform-dispatched: on ROCm it "
    "goes through the vllm::rocm_unquantized_gemm custom op, which has no CPU "
    "kernel, so the CPU-tensor fallback cannot be exercised there.",
)
def test_fallback_preserves_default_method() -> None:
    x = torch.randn(2, 4)
    weight = torch.randn(3, 4)
    # CPU tensors fail the runtime check, forcing the base-method fallback.
    method = k3_gemm.KimiK3LowLatencyLinearMethod({}, {})

    output = method.apply(SimpleNamespace(weight=weight), x)

    torch.testing.assert_close(output, torch.nn.functional.linear(x, weight))
```

评论区精华

尽管没有直接的 review 讨论, 但 PR body 中的分析堪称典范, 详细解释了每个测试失败的技术原因和修复逻辑。例如, 对于 `test_fallback_preserves_default_method`, 作者解释道:
“The premise is CUDA-specific, so the test is gated on `current_platform.is_cuda()` and

its body is unchanged. The method under test lives in `vllm/models/kimi_k3/nvidia/` and is not installed on ROCm, so gating costs no real coverage.”（前提是 CUDA 特有的，因此测试门控在 `current_platform.is_cuda()` 上，其主体保持不变。被测方法位于 `vllm/models/kimi_k3/nvidia/` 中，在 ROCm 上未安装，因此门控不会造成实际覆盖率损失）。这种清晰、具体的分析是有效的技术沟通。

- 测试平台假设的显式处理 (testing): 采用最符合项目惯例的修复方式：对于可选依赖，使用 `pytest.importorskip`；对于平台特有逻辑，使用 `@pytest.mark.skipif` 结合 `current_platform.is_cuda()` 进行门控。这两种方式都能让测试在失败时给出清晰的跳过原因，而不是抛出令人困惑的错误。

风险与影响

- 风险：风险较低，主要为：
 1. 测试覆盖缺失：第二个测试 (`test_fallback_preserves_default_method`) 在 ROCm 上被完全跳过。尽管作者指出被测方法在 ROCm 上不可用，但这意味着其平台分发逻辑中的回退路径在 ROCm 上没有通过此测试进行验证。如果未来该模块被移植到 ROCm，需要同步启用此测试。
 2. 依赖处理模式不一致：第一个测试使用 `pytest.importorskip`，而其他测试（如文件中已有的）可能直接导入。这不影响功能，但可能导致测试失败时的跳过原因信息略有不同。
 - 影响：直接影响：修复了 ROCm CI 作业的失败，提高了 CI 稳定性和可信度。间接影响：
- 用户体验（开发者）：在 ROCm 和 CUDA 两种硬件环境下运行测试套件的开发者，不再会因这些测试而遇到意外的失败。
- 系统可靠性：确保了测试套件在混合平台 CI 中能正确反映实际可测试的功能，避免了因环境假设导致的误报。
- 维护性：提供了清晰的测试门控模式 (`importorskip` 和 `skipif`)，可为未来类似跨平台测试问题提供参考。
- 风险标记：测试覆盖缺口，跨平台依赖处理

关联脉络

- PR #50519 Extend ROCm CI to portable parts of upstream CUDA test groups: 本 PR 修复的测试失败正是由于 PR #50519 扩展了 ROCm CI 覆盖范围后暴露出来的。两者是紧密关联的 CI 维护工作。