

PR #49513 完整报告

vllm-project/vllm

[CI] Use explicit devices in IR tests

合并时间: 2026-07-24 11:08

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49513>

执行摘要

- 一句话: 修复 IR 测试中设备显式传递问题
- 推荐动作: 建议快速合并。这是一个 CI 稳定性修复, 逻辑清晰, 改动审慎。值得关注的设计决策是将全局状态依赖改为显式参数传递, 这符合良好的测试实践。

功能与动机

CI 测试组 `vLLM IR Tests` 在 ROCm 平台上频繁失败。根本原因是 RMSNorm 和 LayerNorm 测试使用类级别的 `setup_class` 设置默认设备 (GPU), 但 `pytest` 函数作用域的 fixture 在参数化测试前重置了设备状态, 导致后续生成的输入张量落在 CPU 上, 而测试的算子是为 ROCm 编译的, 产生设备不匹配错误。此次变更使每个测试用例都显式指定设备, 不再依赖隐式的默认设备。

实现拆解

1. 移除 `setup_class` 中的全局设备设置: 删除了 `tests/kernels/ir/test_layernorm.py` 中 `TestRMSNorm` 类的 `setup_class` 方法, 该方法曾调用 `torch.set_default_device(current_platform.device_type)`。
2. 修改输入生成函数签名: 在 `vllm/ir/ops/layernorm.py` 中, 为 `_rms_norm_input_generator` 和 `_fused_add_rms_norm_input_generator` 函数增加了可选的 `device` 参数 (默认 `None`) 。
3. 更新所有测试调用点: 修改了 `test_layernorm.py` 中所有调用 `generate_inputs` 的测试函数 (包括 `test_native_semantics`, `test_impls`, `test_torch_opcheck`, `test_iter_rejects_unsupported_dtypes`, `test_vllm_c_rms_norm_accepts_nd_input`) , 在每个调用中显式传入 `device=current_platform.device_type`。
4. 清理冗余的 `torch.set_default_device` 调用: 移除了 `test_iter_rejects_unsupported_dtypes` 和 `test_vllm_c_rms_norm_accepts_nd_input` 中残留的 `torch.set_default_device` 语句, 这些语句因全局状态而不再安全, 且已被显式设备参数取代。

关键文件:

- `tests/kernels/ir/test_layernorm.py` (模块 测试覆盖; 类别 `test`; 类型 `test-coverage`; 符号 `setup_class`, `test_native_semantics`, `test_impls`, `test_torch_opcheck`): 核心测试文件, 移除了依赖全局设备状态的 `setup_class`, 并在每个测试方法中显式传递 `device` 参数

，修复了 ROCm CI 失败的根本原因。

- `vllm/ir/ops/layernorm.py` (模块 IR 算子; 类别 `infra`; 类型 `infrastructure`; 符号 `_rms_norm_input_generator`, `_fused_add_rms_norm_input_generator`) : 修改了输入生成器函数, 新增 `device` 参数, 使生成器能够将张量创建在指定设备上, 支持测试端的显式设备传递。

关键符号: `_rms_norm_input_generator`, `_fused_add_rms_norm_input_generator`,
`test_native_semantics`, `test_impls`, `test_torch_opcheck`,
`test_aiter_rejects_unsupported_dtypes`, `test_vllm_c_rms_norm_accepts_nd_input`

关键源码片段

`tests/kernels/ir/test_layernorm.py`

核心测试文件, 移除了依赖全局设备状态的 `setup_class`, 并在每个测试方法中显式传递 `device` 参数, 修复了 ROCm CI 失败的根本原因。

```
# before: class-level setup_class sets default device globally,
# but pytest fixtures may reset it, causing inputs on CPU
# after: every test explicitly passes device to generate_inputs

@pytest.mark.parametrize("dtype", [torch.float16, torch.bfloat16, torch.float32])
@pytest.mark.parametrize("n_tokens", NUM_TOKENS)
@pytest.mark.parametrize("hidden_size", COMMON_HIDDEN_SIZES)
@pytest.mark.parametrize("epsilon", [1e-6, 1e-5])
@pytest.mark.skipif(
    not current_platform.is_cuda_alike() and not current_platform.is_xpu(),
    reason="Currently only kernels on CUDA, ROCm and XPU",
)
class TestRMSNorm:
    # setup_class removed — no more torch.set_default_device

    def test_native_semantics(self, dtype, n_tokens, hidden_size, epsilon):
        # Pass device explicitly instead of relying on global default
        x, weight, epsilon = ir.ops.rms_norm.generate_inputs(
            num_tokens=4,
            hidden_size=8,
            dtype=dtype,
            epsilon=epsilon,
            device=current_platform.device_type,
        )
        out = rms_norm_native(x, weight, epsilon=epsilon)
        # ... rest of test unchanged

    def test_impls(self, dtype, n_tokens, hidden_size, epsilon, provider):
        impl = ir.ops.rms_norm.impls[provider]
        x, weight, eps = ir.ops.rms_norm.generate_inputs(
            num_tokens=n_tokens,
            hidden_size=hidden_size,
```

```
        dtype=dtype,
        epsilon=epsilon,
        device=current_platform.device_type,
    )
    # ... rest of test unchanged
```

vllm/ir/ops/layernorm.py

修改了输入生成器函数，新增 `device` 参数，使生成器能够将张量创建在指定设备上，支持测试端的显式设备传递。

```
@rms_norm.register_input_generator
def _rms_norm_input_generator(
    num_tokens: int,
    hidden_size: int,
    dtype: torch.dtype,
    epsilon: float = 1e-5,
    device: torch.device | str | None = None, # new: explicit device
) -> tuple:
    # Use device if provided; fallback to current default (CPU) if None
    x = torch.randn(num_tokens, hidden_size, dtype=dtype, device=device)
    weight = torch.randn(hidden_size, dtype=dtype, device=device)
    return x, weight, epsilon
```

评论区精华

无人工 review 评论，只有 `claude[bot]` 自动回复提示从 fork 提交需手动触发 review，以及 `mgoin` 的批准。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅限于测试基础设施，不涉及核心逻辑。风险在于：
 - 如果 `current_platform.device_type` 返回意外值，会导致测试在错误设备上运行。但该函数在现有代码中已广泛使用，风险可控。
 - 新的 `device` 参数默认 `None`，若其他地方的调用未更新，`torch.randn` 会退回到 CPU，不会崩溃。
 - 影响：影响范围有限。仅影响 `tests/kernels/ir/test_layernorm.py` 和 `vllm/ir/ops/layernorm.py` 两个文件。对用户无直接影响；对团队而言，修复了 CI 中 ROCm 平台上的 IR 测试失败，提高了 CI 稳定性。
- 风险标记：测试基础设施变更，低风险

关联脉络

- 暂无明显关联 PR