

PR #49512 完整报告

vllm-project/vllm

[CI] Use explicit devices in quantization tests

合并时间: 2026-07-24 00:54

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49512>

执行摘要

- 一句话: 量化测试显式指定设备, 消除 fixture 副作用
- 推荐动作: 可以直接合并。该 PR 虽小, 但为解决 CI 不稳定提供了正确且简洁的方案, 值得学习其对待全局 fixture 影响测试隔离性的处理方式。

功能与动机

Kernels Quantization Test 1 和 2 依赖全局 default device 状态, 但 module-scope fixture 在参数化测试间重置导致 CPU 张量传入 GPU kernel, 造成测试不稳定 (见 Buildkite 构建 #11147)。

实现拆解

1. 删除 module-scope fixture: 在 test_int8_kernel.py 和 test_block_int8.py 中移除 setup_cuda fixture, 该 fixture 通过 torch.set_default_device("cuda") 修改全局状态。
2. 显式指定设备创建张量: 在每个测试函数开头通过 device = current_platform.device_type 获取设备, 所有 torch.rand、torch.randn 等调用添加 device=device 参数。
3. 保留参数化覆盖: 测试参数化组合不变, 仅改动张量创建方式, 不影响测试语义。

关键文件:

- tests/kernels/quantization/test_int8_kernel.py (模块 量化测试; 类别 test; 类型 test-coverage; 符号 setup_cuda): 移除 module-scope fixture, 测试张量显式指定设备, 是解决测试不稳定的主要变更。
- tests/kernels/quantization/test_block_int8.py (模块 量化测试; 类别 test; 类型 test-coverage; 符号 setup_cuda): 同样移除 fixture 并用显式设备替代张量创建, 与 test_int8_kernel.py 对称修复。

关键符号: setup_cuda

关键源码片段

tests/kernels/quantization/test_int8_kernel.py

移除 module-scope fixture, 测试张量显式指定设备, 是解决测试不稳定的主要变更。

```
# 不再依赖 module-scope fixture, 改为在测试函数内显式指定设备
```

```
DYPES = [torch.half, torch.bfloat16]
M = [1, 33]
N = [128, 1024]
K = [256, 4096]
E = [8]
TOP_KS = [2, 6]
SEEDS = [0]
```

```
@pytest.mark.parametrize(
    "M, N, K, E, topk, dtype, seed",
    itertools.product(M, N, K, E, TOP_KS, DYPES, SEEDS),
)
@torch.inference_mode()
def test_w8a8_fp8_fused_moe(default_vllm_config, M, N, K, E, topk, dtype, seed):
    torch.manual_seed(seed)
    device = current_platform.device_type # 获取当前设备类型, 不依赖全局默认设备
    # 所有张量创建均指定 device, 确保与 kernel 运行设备一致
    a = torch.randn((M, K), dtype=dtype, device=device) / 10
    w1_fp32 = (torch.rand((E, 2 * N, K), dtype=torch.float32, device=device) - 0.5) * 2
    w2_fp32 = (torch.rand((E, K, N), dtype=torch.float32, device=device) - 0.5) * 2
    # ... 其余逻辑不变
```

tests/kernels/quantization/test_block_int8.py

同样移除 fixture 并用显式设备替代张量创建, 与 test_int8_kernel.py 对称修复。

删除 module-scope fixture, 改为局部设备指定, 避免 fixture 重置

```
@pytest.mark.parametrize(
    "M,N,K,block_size,out_dtype,seed",
    itertools.product(M, N, K, BLOCK_SIZE, DYPES, SEEDS),
)
@torch.inference_mode()
def test_w8a8_block_int8_matmul(M, N, K, block_size, out_dtype, seed):
    torch.manual_seed(seed)
    device = current_platform.device_type # 动态获取设备, 不依赖全局 default_device
    factor_for_scale = 1e-2
    int8_info = torch.iinfo(torch.int8)
    int8_max, int8_min = int8_info.max, int8_info.min
    # 所有张量都创建在正确 device 上, 不再受 fixture 副作用影响
    A_fp32 = torch.rand(M, K, dtype=torch.float32, device=device)
    A_fp32 = (A_fp32 - 0.5) * 2 * int8_max
    A_fp8 = A_fp32.clamp(min=int8_min, max=int8_max).to(torch.float8_e4m3fn)
    # 其余张量类似, 均显式指定 device
```

评论区精华

无 review 讨论。Claude bot 自动评论但因 fork 跳过, mgoin 直接 approve。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。仅修改测试辅助代码和测试函数内张量创建方式，不涉及生产代码；改动逻辑直观，回归风险小。
- 影响：
 - 用户：无影响。
 - 系统：提高量化 kernel 测试在 AMD CI 的稳定性，消除偶发失败。
 - 团队：减少因测试不稳定导致的 CI 重跑。
 - 风险标记：测试隔离改进

关联脉络

- PR #49609 [CI][Bugfix] Fix test isolation in block_int8/ptpc_fp8 MoE kernel tests: 同一测试隔离主题，修复 MoE 核测试中 fixture scope 导致测试隔离失败。