

PR #45694 完整报告

vllm-project/vllm

[Misc] Add and enable Triton kernel unit tests on XPU

合并时间: 2026-08-07 17:53

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45694>

执行摘要

- 一句话: XPU 启用 5 个 Triton 内核单测并新增 round_int8 覆盖
- 推荐动作: 建议测试与多后端支持方向的工程师精读本 PR: 它直观展示了 "Adjusted" 型 PR 的标准化写法 (skip 条件 + 设备变量 + 参考实现对比), 也是 RFC #48480 系列的可复制模板。若关注 XPU 测试覆盖, 可进一步跟踪后续把 ROCm 也纳入 DEVICE 选择逻辑、并在 XPU CI 中显式纳入这些测试文件的 PR。

功能与动机

RFC #48480 指出 vLLM 的 Triton 内核通过同一 Triton 路径即可在非 NVIDIA 后端 (如 Intel XPU) 上运行, 但对应单测被 CUDA 硬编码 (device='cuda'、get_device_capability() 等) 所限制, 导致纯 Triton 内核在 XPU 上要么报 CUDA 专属错误、要么被直接跳过, 没有获得测试覆盖。该 RFC 提出的 PR 形态之一就是 "Adjusted": 把已有 CUDA 绑定测试改为硬件无关, 并在 PR 描述中说明覆盖的内核与理由。本 PR 正是这一模式的执行, 同时补上了此前没有独立测试的 round_int8 内核 (仅间接覆盖)。

实现拆解

本 PR 完全局限于测试目录, 分五步完成:

1. 改造 INT8 相关测试的模块级 skip 条件 (tests/kernels/quantization/test_block_int8.py、tests/kernels/quantization/test_int8_kernel.py): 把 current_platform.get_device_capability() < (7, 0) 替换为“先判断 is_cuda_alike() or is_xpu(), 不满足则跳过; 再对 CUDA-alike 单独用 has_device_capability((7, 0)) 检查”, 既让 XPU 进入测试路径, 又保证 CUDA 的 skip 语义不变。
2. 设备选择硬件无关化 (tests/kernels/test_flatten_layernorm_guard.py、tests/kernels/core/test_fused_rms_norm_gated.py): 引入模块级 DEVICE = "xpu" if current_platform.is_xpu() else "cuda" (后者为 "xpu:0"/"cuda:0"), 统一替换各测试函数里硬编码的 torch.device("cuda:0")。
3. 一行替换打通 scaled_mm 测试 (tests/kernels/quantization/test_triton_scaled_mm.py): device = "cuda" 改为 device = current_platform.device_type, 使 96 个用例在 XPU 上运行 (1 个跳过)。
4. 新增 test_per_token_quant_int8 (test_int8_kernel.py): 参数化 num_tokens/hidden_dim/dtype, 通过生产入口 per_token_quant_int8 直接覆盖

round_int8 Triton 内核；以 native_per_token_group_quant_int8 为参考实现（per-token 量化等价于 group_size = hidden_dim 的 group 量化），断言 scale 误差不超过 1e-3、量化值误差不超过 1 ULP。

5. 验证配套：作者在 Arc Pro B70 (torch 2.12.0+xpu、triton-xpu 3.7.1) 与 H200 上运行 5 个测试文件全部通过，jikunshang 又触发 Buildkite CI #82813 做全量验证。

关键文件：

- tests/kernels/quantization/test_int8_kernel.py (模块 INT8 内核；类别 test；类型 test-coverage；符号 test_per_token_quant_int8, native_w8a8_per_token_matmul, torch_w8a8_per_column_moe)：除了将 skip 条件改为 CUDA-alike/XPU 双轨外，还新增了 test_per_token_quant_int8，首次为 round_int8 Triton 内核提供独立单元测试，是本 PR 中唯一包含新测试逻辑的文件。
- tests/kernels/test_fla_layernorm_guard.py (模块 归一化内核；类别 test；类型 test-coverage；符号 layer_norm_ref, test_layer_norm_fwd_basic, test_layer_norm_fwd_with_gate, test_layer_norm_fwd_with_groups)：将 8 处硬编码 cuda:0 统一替换为平台选择的 DEVICE 变量，是 XPU 覆盖范围最大的文件（1553 个用例），展示了 DEVICE 变量的标准写法。
- tests/kernels/core/test_fused_rms_norm_gated.py (模块 融合归一化；类别 test；类型 test-coverage；符号 test_compiled_vs_eager, test_compiled_vs_eager_multidim)：覆盖 FusedRMSNormGated 的 forward_cuda 与 torch.compile 分解路径对比，DEVICE 替换使其在 XPU 上验证 torch.compile 行为。
- tests/kernels/quantization/test_block_int8.py (模块 Block INT8；类别 test；类型 test-coverage；符号 test_w8a8_block_int8_matmul)：w8a8 block INT8 量化矩阵乘内核测试的 skip 条件与 test_int8_kernel.py 同步改造，是 RFC 中 "Adjusted" 形态的典型代表。
- tests/kernels/quantization/test_triton_scaled_mm.py (模块 Scaled MM；类别 test；类型 test-coverage；符号 torch_scaled_mm, get_8bit_types)：最小改动（1 行）把模块级 device 从硬编码 cuda 改为 current_platform.device_type，让 96 个 scaled_mm 用例在 XPU 上运行，是硬件无关化的最简示范。

关键符号：test_per_token_quant_int8, test_w8a8_block_int8_matmul, test_layer_norm_fwd_basic, test_layer_norm_fwd_with_gate, test_layer_norm_fwd_with_groups, test_layer_norm_rows_per_block, test_strided_input, test_output_buffer_provided, test_multidimensional_input, test_rmsnorm_gated_forward_native_dtype, test_compiled_vs_eager, test_compiled_vs_eager_multidim

关键源码片段

tests/kernels/quantization/test_int8_kernel.py

除了将 skip 条件改为 CUDA-alike/XPU 双轨外，还新增了 test_per_token_quant_int8，首次为 round_int8 Triton 内核提供独立单元测试，是本 PR 中唯一包含新测试逻辑的文件。

```
# tests/kernels/quantization/test_int8_kernel.py
```

```

# 模块级 skip: 先判断是否 CUDA-alike 或 XPU, 再对 CUDA-alike 单独检查算力。
# 这样 XPU 不再被 CUDA 专属的 get_device_capability() 挡在门外。
from vllm.platforms import current_platform

if not (current_platform.is_cuda_alike() or current_platform.is_xpu()):
    pytest.skip(
        "INT8 Triton kernels require a CUDA-alike or XPU device",
        allow_module_level=True,
    )

if current_platform.is_cuda_alike() and not current_platform.has_device_capability(
    (7, 0)
):
    pytest.skip("INT8 Triton requires CUDA 7.0 or higher", allow_module_level=True)

# 新增的独立单测: 直接通过生产入口 per_token_quant_int8 覆盖 round_int8 内核。
@pytest.mark.parametrize("num_tokens", [1, 33, 64])
@pytest.mark.parametrize("hidden_dim", [128, 1024])
@pytest.mark.parametrize("dtype", DTYPES)
@torch.inference_mode()
def test_per_token_quant_int8(num_tokens, hidden_dim, dtype):
    """Covers the round_int8 Triton kernel via per_token_quant_int8."""
    torch.manual_seed(0)
    device = current_platform.device_type
    x = torch.randn((num_tokens, hidden_dim), dtype=dtype, device=device) * 8

    q, s = per_token_quant_int8(x)
    # Per-token int8 quant 等价于 group_size = hidden_dim 的 per-token-group quant,
    # 因此用 native 实现做参考, 而不是自己重写一遍量化逻辑。
    ref_q, ref_s = native_per_token_group_quant_int8(x, hidden_dim)

    assert q.dtype == torch.int8
    assert q.shape == (num_tokens, hidden_dim)
    assert s.shape == (num_tokens, 1)
    # scale 允许 1e-3 误差; 量化值允许 1 ULP 舍入差, 避免对浮点舍入过度敏感。
    assert torch.allclose(s.float(), ref_s.float(), atol=1e-3, rtol=1e-3)
    assert torch.all((q.float() - ref_q.float()).abs() <= 1)

```

tests/kernels/test_flatten_layernorm_guard.py

将 8 处硬编码 cuda:0 统一替换为平台选择的 DEVICE 变量, 是 XPU 覆盖范围最大的文件 (1553 个用例), 展示了 DEVICE 变量的标准写法。

```

# tests/kernels/test_flatten_layernorm_guard.py
# 统一的设备选择: XPU 上用 xpu, 其余默认 CUDA。
# 注意这里并未覆盖 ROCm, 说明硬件无关化是分步进行的。
from vllm.platforms import current_platform

DEVICE = "xpu" if current_platform.is_xpu() else "cuda"

```

```

def test_layer_norm_fwd_basic(
    num_tokens: int,
    hidden_size: int,
    dtype: torch.dtype,
    seed: int,
    is_rms_norm: bool,
) -> None:
    """Test basic layer norm forward pass without z (gate) tensor."""
    set_random_seed(seed)
    device = torch.device(DEVICE) # 原来是硬编码 torch.device("cuda:0")

    # 输入、权重、偏置全部建在目标设备上, kernel 与参考实现走同一条设备路径
    x = torch.randn(num_tokens, hidden_size, dtype=dtype, device=device)
    weight = torch.randn(hidden_size, dtype=dtype, device=device)
    bias = None if is_rms_norm else torch.randn(hidden_size, dtype=dtype, device=device)
    eps = 1e-6

    out, mean, rstd = layer_norm_fwd(
        x, weight, bias, eps, z=None, is_rms_norm=is_rms_norm
    )
    ref_out = layer_norm_ref(x, weight, bias, z=None, eps=eps, is_rms_norm=is_rms_norm)

    torch.testing.assert_close(out, ref_out, atol=1e-2, rtol=1e-2)

```

评论区精华

评审讨论较少且均为流程性问题：

- mergify[bot] 两次报告合并冲突，要求 rebase；作者推送新 commit 后解决。
- mergify[bot] 报告 pre-commit 失败，mgoin 要求修复 precommit 并关注 Intel CI。
- mayuyuace 给出 LGTM，并请 jikunshang 复核；jikunshang 回复 /ci run 触发 Buildkite CI #82813。
- mgoin 与 jikunshang 最终均 APPROVED。没有留下关于实现细节的技术讨论。
- pre-commit 失败与 Intel CI 关注 (other): 作者后续推送修复提交解决 pre-commit；Intel CI 的显式集成未在本 PR 内完成，由 jikunshang 以 /ci run 触发 Buildkite CI 验证。
- 合并冲突需要 rebase (other): 作者通过后续提交解决冲突，PR 最终可合并。
- 评审复核与 CI 触发 (other): mgoin 与 jikunshang 均 APPROVED，PR 合并。

风险与影响

- 风险：风险较低，但仍需注意几点：
 - 测试语义变化风险：test_flan_layernorm_guard.py 与 test_fused_rms_norm_gated.py 的 DEVICE 逻辑是 is_xpu() else "cuda"，非 XPU 平台（如 ROCm）仍会用 CUDA 设备，硬件无关化尚未完成；不过这不在本 PR 目标内。
 - XPU 数值精度风险：新增的 test_per_token_quant_int8 断言 scale 误差 1e-3、量化值误差 1 ULP，已在 Arc Pro B70 上通过，但不同 XPU 型号或 triton-xpu 版本可能带来

舍入差异，属于单测层面风险，不涉及生产行为。

- CI 集成盲区：本 PR 未修改任何 .buildkite 配置，Intel XPU CI job 是否实际收集这 5 个测试文件取决于既有测试发现逻辑；mgoin 评论也提到要 look into intel ci，意味着 CI 集成可能需要后续 PR 跟进。
- 影响：影响范围为测试基础设施与多后端开发体验：
 - 对 CUDA 用户 / 行为零影响：所有改动仅是设备选择与 skip 条件重构，CUDA 分支保持原路径。
 - 对 Intel XPU 用户：5 个内核测试（INT8 block 量化、INT8 量化、triton_scaled_mm、FLA layernorm guard、FusedRMSNormGated）首次在 XPU 上获得直接验证，能更早暴露 Triton 内核在 XPU 上的回归；新增的 round_int8 独立测试让量化内核回归定位更精准。
 - 对团队：这是 RFC #48480 系列的一个典型落地样例，为后续 "New" 型测试（为无覆盖内核补独立测试）铺路，并示范了 current_platform 的推荐用法。
 - 风险标记：仅测试变更，ROCm 设备选择未纳入，XPU 数值精度依赖，CI 集成未显式覆盖

关联脉络

- PR #48480 [RFC]: Make Triton kernel unit tests hardware-agnostic and cover untested kernels: 本 PR 是 RFC #48480 的落地实现之一，PR body 明确声明 "Part of RFC #48480"，采用其提出的 "Adjusted" PR 形状，并为 round_int8 内核补充 "New" 型独立测试。
- PR #51365 [XPU] quick fix online quantization UT break: 同一 XPU 量化测试修复脉络；该 PR 修复了在线量化单测在 XPU 上的硬编码 cuda 问题，与本 PR 的硬件无关化目标一致，说明 XPU 测试适配是持续进行的工作。
- PR #50393 [Bugfix][Platform] Stop re-initializing NVML on every device-capability check (fixes #50381): 本 PR 将 get_device_capability() 调用替换为 has_device_capability()，与平台层设备能力检查 API 的演进相关；该 PR 表明 current_platform 的能力检查 API 本身也在优化，二者共同影响平台层用法。