

PR #51672 完整报告

vllm-project/vllm

[Misc] Enable test_fused_moe_wn16 on XPU

合并时间: 2026-08-10 21:57

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51672>

执行摘要

- 一句话: XPU 启用 WNA16 MoE 测试, 设备选择改由平台常量驱动
- 推荐动作: 值得快速浏览作为跨平台测试改造的参考模板: 用模块级 `DEVICE_TYPE` 替代硬编码 `device`, 并用 `skipif` 显式表达平台能力边界。无需深入阅读, 因为不涉及生产逻辑。

功能与动机

PR body 明确指出: 原测试硬编码 `device="cuda"`, 因此无法在其他 GPU 平台 (如 XPU) 上运行, 而 `test_fused_moe_wn16` 覆盖的 `fused_moe_kernel_gptq_awq` Triton kernel 需要跨平台验证。同时该 PR 是 #45809 的替代版本, 因为旧分支名含括号被 Buildkite 分支名校验器拒绝 (Invalid branch name. Contains disallowed characters.), 无法运行 CI。

实现拆解

1. 设备常量替换: 在 `tests/kernels/moe/test_moe.py` 中, 将 `test_fused_moe_wn16` 内所有 `device="cuda"` 替换为模块级常量 `DEVICE_TYPE` (即 `current_platform.device_type`), 覆盖输入张量、量化权重 `/scale/zero-point` 张量及 `ep_size>1` 时的索引张量分配。
2. 新增跳过守卫: 在测试函数上添加 `@pytest.mark.skipif(not (current_platform.is_cuda_alike() or current_platform.is_xpu()))`, 使缺少对应 Triton WNA16 kernel 的平台直接跳过, 避免因设备不可用导致失败。
3. 格式整理: 顺带将多行 `torch.empty` 调用按行展开, 提升可读性, 无逻辑变更。
4. 验证配套: 作者在 Intel Arc Pro B70 (torch 2.12.0+xpu, Triton 3.8.0) 上运行 `pytest tests/kernels/moe/test_moe.py -k wn16`, 384 个用例全部通过, 耗时 176s。

关键文件:

- `tests/kernels/moe/test_moe.py` (模块 测试用例; 类别 `test`; 类型 `test-coverage`; 符号 `test_fused_moe_wn16`): 唯一变更文件: 将 `test_fused_moe_wn16` 的设备选择从硬编码 `cuda` 改为 `DEVICE_TYPE`, 并新增 `skipif` 守卫, 使 GPTQ/AWQ WNA16 fused MoE kernel 测试可在 XPU 上运行。

关键符号: `test_fused_moe_wn16`

关键源码片段

[tests/kernels/moe/test_moe.py](#)

唯一变更文件：将 test_fused_moe_wn16 的设备选择从硬编码 cuda 改为 DEVICE_TYPE, 并新增 skipif 守卫，使 GPTQ/AWQ WNA16 fused MoE kernel 测试可在 XPU 上运行。

```
# tests/kernels/moe/test_moe.py
# 模块内已有的平台设备常量：CUDA/ROCm 为 "cuda", XPU 为 "xpu"
DEVICE_TYPE = current_platform.device_type

@pytest.mark.parametrize("m,n,k", FUSED_MOE_WN16_MNK_FACTORS)
@pytest.mark.parametrize("e", NUM_EXPERTS)
@pytest.mark.parametrize("topk", TOP_KS)
@pytest.mark.parametrize("ep_size", EP_SIZE)
@pytest.mark.parametrize("dtype", [torch.bfloat16])
@pytest.mark.parametrize("group_size", [64, 128])
@pytest.mark.parametrize("has_zp", [True, False])
@pytest.mark.parametrize("weight_bits", [4, 8])
# 跳过守卫：没有对应 WNA16 Triton kernel 的平台直接跳过，而非因设备不存在而失败
@pytest.mark.skipif(
    not (current_platform.is_cuda_alike() or current_platform.is_xpu()),
    reason="MoE WNA16 Triton kernels require CUDA/ROCm or XPU.",
)
def test_fused_moe_wn16(m, n, k, e, topk, ep_size, dtype, group_size, has_zp, weight_bits):
    # 输入与参数量化张量统一使用 DEVICE_TYPE，替代原先硬编码的 "cuda"
    a = torch.randn((m, k), device=DEVICE_TYPE, dtype=dtype) / 10
    w1 = torch.randn((e, 2 * n, k), device=DEVICE_TYPE, dtype=dtype) / 10
    w2 = torch.randn((e, k, n), device=DEVICE_TYPE, dtype=dtype) / 10
    score = torch.randn((m, e), device=DEVICE_TYPE, dtype=dtype)

    # 按 weight_bits 决定打包因子与量化标量类型（含 zero-point 与否）
    if weight_bits == 4:
        pack_factor = 2
        quant_type = scalar_types.uint4 if has_zp else scalar_types.uint4b8
    elif weight_bits == 8:
        pack_factor = 1
        quant_type = scalar_types.uint8 if has_zp else scalar_types.uint8b128

    # 量化后的 qweight / scales / qzeros 容器同样落在 DEVICE_TYPE 上
    w1_qweight = torch.empty(
        (e, 2 * n, k // pack_factor), device=DEVICE_TYPE, dtype=torch.uint8
    )
    w2_qweight = torch.empty(
        (e, k, n // pack_factor), device=DEVICE_TYPE, dtype=torch.uint8
    )
    w1_scales = torch.empty(
        (e, 2 * n, k // group_size), device=DEVICE_TYPE, dtype=dtype
    )
    w2_scales = torch.empty((e, k, n // group_size), device=DEVICE_TYPE, dtype=dtype)

    # ep_size > 1 时的专家切分索引也需与权重同设备
```

```
if ep_size > 1:
    local_e = e // ep_size
    e_ids = torch.randint(0, e, (local_e,), device=DEVICE_TYPE, dtype=torch.int32)
    e_map = torch.full((e,), -1, device=DEVICE_TYPE, dtype=torch.int32)
    e_map[e_ids] = torch.arange(local_e, device=DEVICE_TYPE, dtype=torch.int32)
else:
    e_map = None
```

评论区精华

该 PR 无实质代码讨论：claude[bot] 指出 PR 来自 fork，自动化审查默认禁用，需维护者手动触发；维护者 jikunshang 直接批准并触发 Buildkite CI #83139。

- fork 分支的自动化审查禁用 (other): 维护者 jikunshang 直接批准并触发 Buildkite CI #83139，无进一步代码讨论。

风险与影响

- 风险：
 1. CUDA/ROCm 行为回归风险：DEVICE_TYPE 在 CUDA 与 ROCm 上均为 "cuda"，行为等价，作者已显式说明，风险极低。
 2. XPU kernel 覆盖盲点：skipif 允许在无 kernel 平台跳过，属于预期行为，但可能掩盖部分平台 kernel 缺失问题；此风险由其他 kernel 兼容性测试兜底。
 3. 测试范围扩大带来的 CI 时长：XPU 上新增 384 个参数化用例（约 176 秒），可能小幅增加 Intel GPU CI 耗时，但属于一次性成本。- 影响：影响范围限于测试层：为 Intel GPU (XPU) CI 增加 WNA16 GPTQ/AWQ fused MoE kernel 的回归覆盖，同时保持 CUDA/ROCm 原有行为不变。对用户和系统无运行时影响；对团队的收益是让 XPU 平台能持续验证 fused_moe_kernel_gptq_awq kernel 的正确性，填补了该 kernel 在非 CUDA 平台上的测试空白。- 风险标记：测试设备选择变更，新增平台跳过逻辑，仅测试改动无生产代码

关联脉络

- PR #45809 [Misc] Enable test_fused_moe_wn16 on XPU (原版)：本 PR 是 #45809 的替代版本，diff 相同，仅因旧分支名含括号被 Buildkite 校验拒绝而重开。
- PR #51148 [CPU] Enable GPTQ and AWQ quantization for s390x: 同为 GPTQ/AWQ WNA16 量化 kernel 的平台适配，共享 fused_moe_kernel_gptq_awq 相关技术栈，本 PR 是对 XPU 侧测试覆盖的补充。
- PR #47205 [Kernel][XPU] Tensor-descriptor operand loads for Triton W8A8 scaled_mm: 同为 XPU 上 Triton 量化 kernel 的适配工作，体现 Intel GPU 上量化 kernel 持续获得支持与验证的趋势。