PR #46434 完整报告

vllm-project/vllm

[ROCm][CI] Enable modular OAI Triton MoE tests

合并时间：2026-08-20 02:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46434

执行摘要

- 一句话：为 ROCm 启用 OAI Triton MoE 测试并修复布局
- 推荐动作：

功能与动机

实现拆解

1. 测试与验证 - tests/kernels/moe/test_modular_oai_triton_moe.py（test-coverage）：测试配套；包含 测试覆盖调整、控制流调整、配置键调整；+9/-8

关键文件：

- tests/kernels/moe/test_modular_oai_triton_moe.py（模块 测试；类别 test；类型 test-coverage）：唯一变更文件，决定测试在哪些设备上运行并修正 ROCm 上的 MXFP4 布局假设。

关键符号：未识别

关键源码片段

tests/kernels/moe/test_modular_oai_triton_moe.py

唯一变更文件，决定测试在哪些设备上运行并修正 ROCm 上的 MXFP4 布局假设。

```
# tests/kernels/moe/test_modular_oai_triton_moe.py

# 将原来的平台判断（is_cuda_alike/is_cuda）改为后端能力判断，
# 使测试能在 OAI Triton MoE 后端实际支持的设备（含 ROCm gfx9/gfx1x）上运行，
# 而不是一刀切地在所有非 CUDA 设备上跳过。

@pytest.mark.skipif(
    not OAITritonExperts._supports_current_device(),
    reason="OAI Triton MoE is not supported on this device.",
)
@pytest.mark.parametrize("dtype", [torch.bfloat16])
def test_oai_triton_moe(...):
    ...
    x = torch.randn((m, k), dtype=dtype, device="cuda")
```

```
    x_tri = F.pad(x, (0, x_pad, 0, 0)) # 按 Triton 后端要求的 padding 对齐输入（MXFP4 布局需要）

    with set_current_vllm_config(VllmConfig()):
        out_ref = torch_moe_impl(x, w1, w2, w1_bias, w2_bias, topk_weights, topk_ids)
        out = oai_triton_moe_impl(
            x_tri, w1_tri, w2_tri, w1_precision_config, w2_precision_config,
            w1_bias_tri, w2_bias_tri, num_experts, topk_weights, topk_ids, unfused,
        )
        out = out[..., :k] # 去掉 padding 维度，只保留原始 k 列，以便与参考实现比较

    assert_close(ref=out_ref, tri=out, maxtol=0.025, rmstol=0.005)


# unfused 用例同样改用后端能力判断，并在进入前先检查 _moe_C::moe_sum 是否可用。
# 某些 ROCm 构建（如 MI355X）不注册该 op，因此仅对 unfused 用例跳过，
# 而 fused 用例仍可运行并验证数值正确性。

@pytest.mark.skipif(
    not UnfusedOAITritonExperts._supports_current_device(),
    reason="Unfused OAI Triton MoE is not supported on this device.",
)
def test_unfused_oai_triton_experts_apply_direct_deepseek_v4_topology(workspace_init):
    ...
    x = torch.randn((m, k), dtype=dtype, device="cuda")
    x_tri = F.pad(x, (0, x_pad, 0, 0)) # 与上方一致，确保 Triton 输入的 padding 对齐
    ...
    experts = UnfusedOAITritonExperts(moe_config, quant_config)
    # 原代码在构造专家实例后用 pytest.skip 跳过，现在改用 skipif 提前声明，
    # 并将 moe_problem_size 的入参从 x 改为 x_tri，保证尺寸计算与后续 apply 一致
    _, _, N, K, top_k = experts.moe_problem_size(x_tri, w1_tri, w2_tri, topk_ids)
    ...
    experts.apply(
        hidden_states=x_tri, # 之前错误地传入了未 padding 的 x，现统一为 x_tri
        ...
    )
    output = output[..., :k] # 裁剪 padding 输出
    assert_close(ref=out_ref, tri=output, maxtol=0.025, rmstol=0.005)
```

评论区精华

- 暂无高价值评论线程

风险与影响

- 风险：
- 影响：
  - 风险标记：测试文件对真实 device 的依赖

关联脉络

- PR #41100 [ROCm][CI] Extended Fused MoE and FP8 MoE test support: 同为 ROCm 下 MoE 测试扩展，涉及同一测试文件 tests/kernels/moe/test_modular_oai_triton_moe.py，且扩展了 ROCm MoE 测试矩阵。