PR #46758 完整报告

vllm-project/vllm

[ROCm][CI TG] refactor and fix deepep_moe test group

合并时间：2026-06-27 10:45

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46758

执行摘要

- 一句话：修复 ROCm DeepEP MoE 测试的 buffer 重用和精度检查
- 推荐动作：值得精读，特别是跨平台精度验证的设计模式（宽松 vs 严格）、测试工具函数提取以及 buffer 重用突破平台限制的策略。关注 check_accuracy 在测试工具箱中的价值。

功能与动机

修复 ROCm CI 上 test_deepep_moe.py 的失败，具体报错是 rocSHMEM 的 Unknown allocator type SingleHeap，因为 DeepEP 低延迟 buffer 在循环中反复创建，而 rocSHMEM 只支持每个进程一次分配。同时，ROCm 的 FP8 格式（e4m3fnuz）与 NVIDIA 的 e4m3fn 舍入不同，导致严格 bit 级比较失败，需要调整精度检查策略。

实现拆解

1. Buffer 重用：在 run_modular_kernel 函数中，将 FusedMoEQuantConfig.make 和 make_modular_kernel 调用移动到循环外部，仅构建一次 kernel 及其 DeepEP buffer。在 process_chunk 中，为低延迟模式增加 clean_low_latency_buffer 调用，以清理上次 chunk 残留的脏数据，确保低延迟 kernel 需要零初始化区域。
2. 精度检查提取：将 test_ocp_mx_moe.py 中的 check_accuracy 函数移动到 tests/kernels/moe/utils.py 作为公共工具函数，该函数通过计算超出 atol+rtol 范围的元素比例来判定通过与否。
3. 新增断言函数：在 test_deepep_moe.py 中定义 assert_deepep_close，内部检查是否使用 FP8 且平台为 ROCm（is_fp8_fnuz()），若是则调用 check_accuracy 使用宽松阈值（atol=rtol=0.15, percent=0.95）；否则使用 torch.testing.assert_close（atol=rtol=0.06）保持严格比较。
4. 测试配置调整：移除了 chunk 级别的 rank_token_scales 切片逻辑，因为 buffer 重用后 kernel 需要全量 scale 参数。

关键文件：

- tests/kernels/moe/test_deepep_moe.py（模块 DeepEP 测试；类别 test；类型 test-coverage；符号 assert_deepep_close, process_chunk）：核心变更文件：实现了 DeepEP buffer 重用机制和平台感知的精度检查函数 assert_deepep_close。
- tests/kernels/moe/test_ocp_mx_moe.py（模块 OCP MX 测试；类别 test；类型 test-coverage；符号 check_accuracy）：移除本地 check_accuracy 定义，改为从公共模

块导入，便于复用。

- tests/kernels/moe/utils.py（模块 MoE 测试辅助；类别 test；类型 test-coverage；符号 check_accuracy）：新增公共 check_accuracy 函数，被两个测试文件使用，提供基于 mismatch 比例的精度验证逻辑。

关键符号：assert_deepep_close, check_accuracy

关键源码片段

tests/kernels/moe/test_deepep_moe.py

核心变更文件：实现了 DeepEP buffer 重用机制和平台感知的精度检查函数 assert_deepep_close。

```
# ===== Buffer reuse: kernel created once before the loop =====
# 之前 kernel 创建在循环内部，导致 rocSHMEM 反复初始化
quant_config = FusedMoEQuantConfig.make(
    q_dtype, w1_scale=w1_scale, w2_scale=w2_scale,
    per_act_token_quant=per_act_token_quant,
    a1_scale=test_tensors.rank_token_scales,
)
mk: FusedMoEKernel = make_modular_kernel(
    pg, pgi, low_latency_mode, hidden_size, dp_size,
    num_experts, num_local_experts, q_dtype,
    use_fp8_dispatch, quant_config,
)

def process_chunk(chunk_start, chunk_end, skip_result_store=False):
    rank_tokens_chunk = test_tensors.rank_tokens[chunk_start:chunk_end]
    topk_weights_chunk = test_tensors.topk_weights[chunk_start:chunk_end]
    topk_chunk = test_tensors.topk[chunk_start:chunk_end]
    if low_latency_mode:
        # 复用 buffer 后需要清理低延迟 buffer 以确保零初始化区域的正
        mk.prepare_finalize.buffer.clean_low_latency_buffer(
            MAX_TOKENS_PER_RANK, hidden_size, num_experts
        )
    out = mk.apply(
        hidden_states=rank_tokens_chunk, w1=w1, w2=w2,
        topk_weights=topk_weights_chunk, topk_ids=topk_chunk,
        activation=MoEActivation.SILU, global_num_experts=num_experts,
        expert_map=build_expert_map(), apply_router_weight_on_input=False,
    )
    if not skip_result_store:
        out_hidden_states[chunk_start:chunk_end, :].copy_(out, non_blocking=True)

# ===== Platform-aware assertion =====
def assert_deepep_close(
    expected: torch.Tensor, actual: torch.Tensor,
    k: int, use_fp8_dispatch: bool,
) -> None:
```

```
if use_fp8_dispatch and current_platform.is_fp8_fnuz():
    # ROCm 的 e4m3fnuz 格式与 NVIDIA e4m3fn 的舍入不同，
    # 即使 kernel 正确也会产生少量 outlier，故允许 5% 不匹配
    check_accuracy(expected, actual, atol=1.5e-1, rtol=1.5e-1, percent=0.95)
    return
# 其他平台（包括 CUDA）保持严格比较，确保精度回归可被捕获
torch.testing.assert_close(expected, actual, atol=6e-2, rtol=6e-2)
```

评论区精华

在 Review 中，tjtanaa 指出需要保留 CUDA 的严格比较逻辑，避免 CUDA 精度回归被宽松阈值掩盖。divakar-amd 按要求在 assert_deepep_close 中添加了平台条件判断，CUDA 路径保持使用 torch.testing.assert_close。该讨论已解决。

- 保留 CUDA 严格比较逻辑 (correctness): divakar-amd 修改代码，添加条件判断，CUDA 路径使用严格比较，ROCm FP8 路径使用宽松 check_accuracy。

风险与影响

- 风险：主要风险包括：1）assert_deepep_close 中的平台检测依赖 current_platform.is_fp8_fnuz()，若未来 ROCm FP8 格式有变或新增平台，需更新此判断。2）clean_low_latency_buffer 的正确性假设：若该清理方法不彻底，可能导致脏数据影响结果。3）check_accuracy 允许 5% 的不匹配，可能掩盖小规模 kernel 回归。但这些风险都限定在测试层，不影响生产逻辑。
- 影响：直接影响 ROCm CI 中 deepep 测试组，使其稳定通过。对 CUDA 无影响（严格比较不变）。对开发者：公共 check_accuracy 可作为 MoE 测试的通用精度验证工具。无用户可见功能变化。
- 风险标记：FP8 精度放宽依赖 ROCm 检测，Buffer 重用依赖 clean_low_latency_buffer 正确性

关联脉络

- PR #46760 [ROCm][Bugfix] Pass num_kv_splits to aiter mla_reduce_v1: 同为 ROCm 平台测试修复，涉及 ROCm 特定 bugfix。
- PR #46859 [Hardware][AMD][CI] Fix Kernels Quantization test timeout: 同为 AMD CI 测试稳定性修复。