PR #46080 完整报告

vllm-project/vllm

[Hardware][AMD][CI] Fix Kernels Attention test groups

合并时间：2026-06-22 06:10

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46080

执行摘要

- 一句话：修复 AMD CI 中 Kernels Attention 测试组
- 推荐动作：建议阅读该 PR 以了解以下实践：
  1. 如何在不同平台间管理数值精度差异（sdpa_kernel 用法）。
  2. FP8 数据类型在混合格式场景下的测试策略（is_extra 参数）。
  3. CI 配置中从试运行（optional: true）转为门禁（optional: false）的演进方式。

该 PR 本身变更简单，但体现了多平台兼容性测试的常见挑战。

功能与动机

PR body 指出目的是修复 gfx942 和 gfx950 上的 Kernels Attention 测试组。这是 PR #45786 的重做，原 PR 因合并冲突无法直接合并。关联的 Issue 评论进一步说明需要更新 test_sparse_attn_decode_ragged_kernel 以匹配 PR #45681 中额外缓存采用 OCP FP8 格式的变更。

实现拆解

1. 调整 DSv4 测试的 FP8 数据类型处理（tests/kernels/attention/test_rocm_triton_attn_dsv4.py）：为 _pack_fp8_ds_mla_cache 和 _read_fp8_ds_mla_cache 函数添加 is_extra 参数，使额外缓存使用 OCP FP8（float8_e4m3fn），主缓存使用平台默认 dtype。
2. 优化 attention 选择器测试的平台检测（tests/kernels/attention/test_attention_selector.py）：将平台导入改为基于 current_platform 的条件判断，并在非因果测试中支持 ROCm 平台。
3. 提升 prefix prefill 测试的数值精度（tests/kernels/attention/test_prefix_prefill.py）：在 ROCm 上强制使用 SDPBackend.MATH 后端以避免半精度数值误差。
4. 简化 unified attention 测试的 FP8 dtype（tests/kernels/attention/test_triton_unified_attention.py）：统一使用 current_platform.fp8_dtype() 消除条件分支。
5. 更新 CI 配置（.buildkite/test_areas/kernels.yaml 和 .buildkite/test-amd.yaml）：调整超时时间、将 MI300 测试设为 gating，并添加 AMD mirror 依赖。

关键文件：

- tests/kernels/attention/test_rocm_triton_attn_dsv4.py（模块 DSv4 测试；类别 test；类型 test-coverage；符号 _pack_fp8_ds_mla_cache, _read_fp8_ds_mla_cache,

_ref_sparse_decode_ragged）：核心测试逻辑调整，涉及 FP8 数据类型区分，是 PR 的主要测试变更。

- tests/kernels/attention/test_attention_selector.py（模块 注意力选择器测试；类别 test；类型 test-coverage）：优化平台检测逻辑，使测试在 ROCm 上正确跳过或运行。
- tests/kernels/attention/test_prefix_prefill.py（模块 前缀预填充测试；类别 test；类型 test-coverage）：在 ROCm 上强制使用 Math SDPA 后端以确保数值精度。
- tests/kernels/attention/test_triton_unified_attention.py（模块 统一注意力测试；类别 test；类型 test-coverage）：简化 FP8 dtype 定义，消除平台条件分支。
- .buildkite/test_areas/kernels.yaml（模块 CI 配置；类别 config；类型 configuration）：为 AMD 添加 mirror 配置，准确保存源文件依赖。
- .buildkite/test-amd.yaml（模块 AMD CI 配置；类别 config；类型 configuration）：调整 MI300/MI355 测试组的超时和可选性，使其成为门禁。

关键符号：_pack_fp8_ds_mla_cache, _read_fp8_ds_mla_cache

关键源码片段

tests/kernels/attention/test_rocm_triton_attn_dsv4.py

核心测试逻辑调整，涉及 FP8 数据类型区分，是 PR 的主要测试变更。

```
def _pack_fp8_ds_mla_cache(
    kv: torch.Tensor,
    block_size: int,
    is_extra: bool = False # 是否是对应额外缓存的 pack；额外缓存使用 OCP FP8 (float8_e4m3fn)，主缓存使用平台默认 dtype
) -> torch.Tensor:
    """将 KV 打包成 DeepSeek V4 的 FP8 MLA 缓存格式。

    在 ROCm 上，平台默认 dtype 为 float8_e4m3fnuz，而额外缓存需统一使用 OCP FP8。
    """
    assert kv.shape[-1] == HEAD_DIM
    num_tokens = kv.shape[0]
    num_blocks = (num_tokens + block_size - 1) // block_size
    cache = torch.zeros(
        (num_blocks, block_size, 584), dtype=torch.uint8, device=kv.device,
    )
    cache_flat = cache.view(torch.uint8).flatten()
    # 主缓存与额外缓存的 NOPE 部分使用不同的 FP8 dtype
    kv_nope_fp8 = (
        kv[:, :NOPE_HEAD_DIM]
        .to(torch.float8_e4m3fn if is_extra else current_platform.fp8_dtype())
        .view(torch.uint8)
    )
    kv_rope_u8 = kv[:, NOPE_HEAD_DIM:].contiguous().view(torch.uint8)
    for slot in range(num_tokens):
        block_idx = slot // block_size
        pos = slot % block_size
```

```
            block_base = block_idx * cache.stride(0)
            token_base = block_base + pos * 576
            scale_base = block_base + block_size * 576 + pos * 8
            cache_flat[token_base: token_base + NOPE_HEAD_DIM].copy_(kv_nope_fp8[slot])
            cache_flat[token_base + NOPE_HEAD_DIM: token_base + NOPE_HEAD_DIM + ROPE_
            HEAD_DIM * 2].copy_(kv_rope_u8[slot])
            cache_flat[scale_base: scale_base + 7].fill_(127) # 默认 scale = 127
        return cache
```

评论区精华

PR 没有 review 评论，但 Issue 评论中 mawong-amd 说明：‘Updated tests/kernels/attention/test_rocm_triton_attn_dsv4.py to bring test_sparse_attn_decode_ragged_kernel in line with the changes in https://github.com/vllm-project/vllm/pull/45681, where the extra cache has data in OCP FP8 format for all platforms.’ 该评论澄清了测试调整的技术背景——混合使用 OCP 和 UZ FP8 格式是设计意图，而非临时修补。因此，is_extra 参数的设计决策得到了合理解释。

- 更新测试匹配 PR#45681 的 FP8 格式变更 (design): 已采纳，添加 is_extra 参数以区分主缓存和额外缓存的 FP8 类型。

风险与影响

- 风险：风险主要集中三个方面：
  - 测试数值精度：在 ROCm 上强制使用 MATH 后端可能掩盖 triton 算子中的精度问题，但该改动仅影响参考实现（ref），而非被测算子，因此风险较低。
  - FP8 dtype 统一：test_triton_unified_attention.py 中统一使用 current_platform.fp8_dtype() 可能使测试在 ROCm 上使用 float8_e4m3fnuz，需要确认与生产环境的默认量化格式一致。
  - CI 配置变更：超时时间大幅缩短（如 180→15 分钟），若环境性能波动可能导致测试超时失败；但 PR 作者已在实际硬件上验证通过，风险可控。
- 影响：影响范围限定在 AMD CI 流程：
  - 测试组：MI300_1: Kernels Attention %N、MI355_1: Kernels Attention %N、MI355_1: Kernels (B200-MI355) 现在可以在 AMD CI 中稳定运行并作为门禁（gating）条件。
  - 开发者：提交涉及 AMD 注意力的代码变更时，将自动触发这些测试，确保回归被早期捕获。
  - 用户：无直接影响，该 PR 不修改任何生产代码。
  - 团队：减少了手动检查测试结果的需求，提升 CI 可靠性。
  - 风险标记：测试数值精度调整，FP8 dtype 统一，CI 超时缩短

关联脉络

- PR #45681 将额外缓存的 FP8 数据统一为 OCP FP8 格式：本 PR 在测试中跟进该 PR 的变更，使测试正确反映生产行为。
- PR #45786 修复 Kernels Attention 测试组（原始 PR）：本 PR 是该 PR 的重做，解决合并冲突。