PR #49309 完整报告

vllm-project/vllm

[ROCm][CI] Use explicit wvSplitKrc skinny-GEMM test tolerance for bf16 (gfx950)

合并时间：2026-07-31 11:25

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49309

执行摘要

- 一句话：修复 gfx950 上 wvSplitKrc bf16 测试容差误报
- 推荐动作：值得快速浏览而非精读。代码改动极小（+8/-3），但 PR body、issue 评论与 review 中关于 bf16 ULP、catastrophic cancellation、Xavier 初始化以及 ULP 距离断言为何不适用的数值分析，对任何编写或维护 kernel 数值测试的工程师都有参考价值。值得关注的决策点：容差放宽必须 scoped 到具体失效分支、以可观测的最坏误差和 finfo(eps) 为上下界、并用 golden standard 交叉验证"不是 kernel bug"。

功能与动机

在 gfx950（MI355X）上，test_rocm_wvsplitkrc_kernel 的 xnorm=True + bf16 + bias 参数化组合在旧断言 atol=1e-3, rtol=1e-8 下失败。作者在 PR body 中论证这是 bounded bf16 rounding 而非 kernel bug：带 bias 时 ref = A·Bᵀ + bias 处于 ~O(1) 量级，bf16 一个可表示步长 2^-8 = 0.00390625 约为旧 atol 的 3.9 倍；2034 个非跳过组合中最坏绝对差恰好为一个 bf16 step（0.00390625），且不随 K 增长、固定 seed 下可复现。同时 review 指出测试所用的 Xavier 初始化不严格正确（均匀分布缩放不满足 Xavier 的均值 / 方差要求），需要一并修正。

实现拆解

1. 修正测试输入分布（Xavier 初始化）：tests/kernels/quantization/test_rocm_skinny_gemms.py 第 153-154 行将 A = (torch.rand(n, k, ...) * 2 - 1) * xavier 改为 A = torch.randn(n, k, ...) * xavier（B 同理），使每个元素独立服从 N(0, xavier^2)，符合 Xavier normal 定义。该改动由 review 中 mawong-amd 的指正驱动，虽然不改变实测 kernel 与参考的对比行为，但使测试前提在数值上成立。
2. 断言容差 scoped 放宽：xnorm=True 分支从固定 atol=1e-3, rtol=1e-8 改为 atol = 5e-3 if (dtype == torch.bfloat16 and BIAS is not None) else 1e-3，rtol 保持 1e-8。5e-3 的选取依据：高于观测最坏差 3.9e-3 留有缓冲，低于 finfo(bf16).eps 约 7.8e-3，避免掩盖一个 ULP 以上的真实回归；fp16 最坏差 4.88e-4 远低于 1e-3，无需放宽。
3. 容差方案的演进：初版直接松绑 bf16（提交 5b533cb）→ 中间版统一用 finfo(dtype).eps（review 认为语义不清，其为 1.0 幅值处的 1 ULP，与本测试 ~O(1) 输出量级的关联不明显）→ 最终改为显式、bias-scoped 的 5e-3（提交 271b9e2）；期间曾尝试将 rtol 放宽至 1e-3，因该检查以 atol 为主导而恢复严格 1e-8（提交 e111b4a）。

4. 验证与配套：纯测试变更，无配置、schema、部署改动。gfx950 上全组合 sweep（bf16 + fp16 各 2034 combos）0 失败；test_rocm_skinny_gemms.py 整组运行 9960 passed / 1080 skipped（约 68 分钟）。CI 由维护者触发并通过（Buildkite build #81381）。

关键文件：

- tests/kernels/quantization/test_rocm_skinny_gemms.py（模块 内核测试；类别 test；类型 test-coverage；符号 test_rocm_wvsplitkrc_kernel）：唯一变更文件，承载两项修复：修正测试输入的 Xavier 初始化（rand 改 randn），以及对 xnorm=True + biased bf16 分支进行 scoped atol 放宽；本 PR 的全部内容都集中在此，直接决定 gfx950 CI 是否通过。

关键符号：test_rocm_wvsplitkrc_kernel

关键源码片段

tests/kernels/quantization/test_rocm_skinny_gemms.py

唯一变更文件，承载两项修复：修正测试输入的 Xavier 初始化（rand 改 randn），以及对 xnorm=True + biased bf16 分支进行 scoped atol 放宽；本 PR 的全部内容都集中在此，直接决定 gfx950 CI 是否通过。

```
# 完整参数空间来自：xnorm、n/k/m、dtype、seed、padded_a、bias_mode 的多维参数化，
# 其中 xnorm、n/k/m 的参数化装饰器在本函数上方（此处省略）
@pytest.mark.parametrize("m", M_FACTORS_WVSPLITKRC)
@pytest.mark.parametrize("dtype", DTYPES)
@pytest.mark.parametrize("seed", SEEDS)
@pytest.mark.parametrize("padded_a", [False, True])
@pytest.mark.parametrize("bias_mode", BIAS_MODES)
@pytest.mark.skipif(not current_platform.is_rocm(), reason="only test for rocm")
@pytest.mark.skipif(not on_gfx950(), reason="only meant for gfx950")
def test_rocm_wvsplitkrc_kernel(xnorm, n, k, m, dtype, seed, padded_a, bias_mode):
    torch.manual_seed(seed)
    cu_count = num_compute_units()

    # 根据 m、k、N 与 CU 数量判断 wvSplitKrc 是否适用，过大的配置直接跳过
    N_p2 = 1 << (n - 1).bit_length()
    rndup_cus = ((m + 64 - 1) // 64) * ((k + 512 - 1) // 512)
    GrpsShrB = min(N_p2 // 16, 4)
    CuNeeded = rndup_cus * GrpsShrB
    fits_wvsplitkrc = (N_p2 * m * ((k + 512 - 1) // 512)) <= 128 * 1024 * 12
    fits_wvsplitkrc &= CuNeeded <= cu_count
    if not fits_wvsplitkrc:
        pytest.skip("Too large for wvSplitKrc")

    # 修正后的 Xavier 初始化：每个元素独立服从 N(0, xavier^2)。
    # 旧写法 (torch.rand * 2 - 1) * xavier 只是均匀分布缩放，方差仅 xavier^2 / 3，
    # 不满足 Xavier normal 的均值 / 方差要求（review 指正后修正）
    xavier = math.sqrt(2 / k) if xnorm else 1
    A = torch.randn(n, k, dtype=dtype, device="cuda") * xavier
    B = torch.randn(m, k, dtype=dtype, device="cuda") * xavier
```

```
    if padded_a:
        A = pad_fp8(A)

    # bias 未归一化、取值在 [-1, 1]，会把 ref = A·Bᵀ + bias 抬到 ~O(1) 量级，
    # 这正是 bf16 舍入边界问题出现的地方
    BIAS = None
    if bias_mode == 1:
        BIAS = torch.rand(m, dtype=dtype, device="cuda") * 2 - 1
    elif bias_mode == 2:
        BIAS = torch.rand(n, m, dtype=dtype, device="cuda") * 2 - 1
    elif bias_mode == 3:
        BIAS = torch.rand(1, m, dtype=dtype, device="cuda") * 2 - 1

    ref_out = torch.nn.functional.linear(A, B, BIAS)
    out = ops.wvSplitKrc(A, B, cu_count, BIAS)

    if xnorm:
        # xnorm 分支以 atol 为主导（rtol = 1e-8 几乎不起作用）。
        # 带 bias 时输出位于 ~O(1)，bf16 一个 ULP 约 3.9e-3 超过旧 atol 1e-3：
        # 只对 biased bf16 放宽到 5e-3（高于观测最坏差 3.9e-3，低于
        # finfo(bf16).eps 约 7.8e-3），其余组合保持 1e-3 严格容差；
        # fp16 的 ULP 约 4.9e-4 远低于 1e-3，无需放宽
        atol = 5e-3 if (dtype == torch.bfloat16 and BIAS is not None) else 1e-3
        torch.testing.assert_close(out, ref_out, atol=atol, rtol=1e-8)
    else:
        torch.testing.assert_close(out, ref_out, atol=1e-3, rtol=1e-2)
```

评论区精华

Review 中最有价值的交锋集中在三点：

1. Xavier 初始化指正：mawong-amd 指出旧写法 (torch.rand(...) * 2 - 1) * xavier 并非 Xavier 初始化——Xavier normal 要求每个元素独立服从均值为 0、方差 2/(n+k) 的正态分布；Xavier uniform 要求 [-r, r]（r = sqrt(6/(n+k))）上的均匀分布。作者采纳并改为 torch.randn(...) * xavier。
2. 分歧根源的交叉验证：mawong-amd 用 float64 计算再舍入到 bf16 的 golden standard 验证，结论是当 golden 输出足够大时，bf16 参考与 kernel 都最多偏离 1 ULP；接近 0 时两者因 FP 减法（正负抵消，catastrophic cancellation）偏离更多 ULP 但幅度相近——即分歧是舍入与抵消的本质结果，不是 kernel bug。
3. 容差方案的取舍：作者在 issue 评论中说明 ULP 距离方案不可行——(ulp > 0) + outlier 预算在 46/2034 组合失败（近零输出 ±1 ULP 数量远超预算），严格 (ulp <= 1).all() 最坏 ULP 距离达 8；且 review 反馈 finfo(bf16).eps 语义不清，最终才选择显式 scoped 的 5e-3。

- Xavier 初始化不严格正确 (correctness): 作者将 A、B 改为 torch.randn(...) * xavier，使每个元素服从 N(0, xavier^2)，符合 Xavier normal 定义。

- bf16 分歧来源与容差放宽的合理性 (correctness): 分歧被刻画为 bf16 舍入边界而非 kernel bug，scoped 放宽 atol 到 5e-3 合理，评审者 approve。
- 为何不采用 ULP 距离断言 (design): 放弃 ULP 方案，改用基于绝对误差观测的显式 scoped atol=5e-3。

风险与影响

- 风险:
  - 容差放宽的回归敏感度（低风险）：放宽严格限定在 biased bf16 的 xnorm 分支，5e-3 仍低于 bf16 eps 约 7.8e-3，且 fp16、非 bias bf16 与 xnorm=False 分支保持 1e-3 严格容差。但该测试从此对 biased bf16 分支绝对误差小于 5e-3 的 kernel 回归不再敏感，需要依赖其他精度分析或 golden-standard 测试兜底。
  - 平台覆盖有限：测试整体 skipif on_gfx950()，本 PR 的数值观测（1 ULP ≈ 3.9e-3、最坏差 0.00390625）仅在 MI355X 上验证；其他 ROCm 平台（如 gfx90a、gfx942）的 bf16 舍入特征未必相同，未来若在这些平台启用该测试需重新评估容差。
  - 输入分布变更的潜在影响：A、B 从均匀分布改为正态分布，改变了数值覆盖域；作者以 2034 combos × 2 dtypes 全量 sweep 验证 0 失败，但长期 flakiness 仍需 CI 观察。
  - 测试成本：该测试组单次全量运行约 68 分钟（9960 用例），是 CI 的持续重负载，本 PR 未改变用例数，但有累积性成本。
- 影响:
  - 用户 / 运行时：无影响，纯测试变更，不涉及 kernel、推理路径或任何生产代码。
  - 系统 /CI：解除 gfx950（MI355X）上的 CI 阻塞，使 wvSplitKrc kernel 的正确性测试在该平台可稳定通过，避免误报失败掩盖真实回归。
  - 团队：为 ROCm 测试确立了“显式 scoped atol + 保持严格 rtol + 注释说明数值依据”的容差调整模式，后续同类问题可复用该决策框架；评审中确立的 double-precision golden standard 对比方法也是可复用的验证手段。
  - 风险标记：测试容差放宽，仅 gfx950 平台验证，重测试组（约 68 分钟），输入分布变更（均匀→正态）

关联脉络

- PR #50450 [ROCm][CI] Use larger atol value for INT3 in test_quick_all_reduce.py: 同一模式：针对 ROCm 平台数值特征调整测试 atol 以修复 CI 失败，本 PR 的容差放宽决策可为后者提供参考；两者共同体现 AMD/CI 侧对 ROCm 测试数值容差的系统性治理。