

PR #51173 完整报告

vllm-project/vllm

[ROCm][CI] Keep rocprofiler-sdk out of DeepEP HT MoE test workers

合并时间: 2026-08-06 04:28

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51173>

执行摘要

- 一句话: ROCm 下 DeepEP MoE 测试禁用 rocprofiler 防段错误
- 推荐动作: 值得精读: 改动虽小, 但根因分析非常扎实——通过最小复现 (纯 HIP + rocprofv3) 把问题定位到 ROCm 运行时, 并设计了明确的回退条件。适合作为“测试与外部工具链缺陷交互”的处理范本, 也提示 CI 维护者在类似场景下优先隔离环境变量而非修改生产逻辑。

功能与动机

PR body 明确说明: 在 ROCm 上, 28 个 high-throughput 版 test_deep_ep_moe 变体全部以 SIGSEGV 失败, 而 low-latency 变体全部通过。崩溃发生在测试体完成后、进程 teardown 阶段, 位于 ROCm HSA runtime 内部——AqlQueue 析构器写入已被 unmap 的信号内存, 该路径只在 rocprofiler-sdk 挂载时才会触达, 而 torch 会隐式挂载 profiler 且 worker 通过 spawn 继承。作者用纯 HIP 程序在 rocprofv3 下复现同样崩溃, 判定缺陷属于 ROCm 并需上报 AMD。

实现拆解

1. 变更入口: 唯一改动文件为 tests/kernels/moe/test_deep_ep_moe.py, 在 test_deep_ep_moe 函数签名中新增 pytest 内置 fixture monkeypatch, 用于保证环境变量在用例结束后自动恢复, 不影响其他测试。
2. 守卫逻辑: 在函数体开头增加 current_platform.is_rocm() 判断, 仅 ROCm 平台执行 monkeypatch.setenv("ROCPROFILER_REGISTER_ENABLED", "0"), 禁用 rocprofiler 注册。
3. 传播机制: DeepEP 测试通过 parallel_launch 生成多进程 worker, 环境变量随 spawn 继承, 因此不用修改 worker 内部代码即可让 profiler 脱离 worker 进程。
4. 范围控制: 只作用于 high-throughput 测试路径, low-latency 变体未改动; 注释中说明这是临时 workaround, 根因已由 ROCm 上游 PR 修复。
5. 配套验证: CI 在 2x MI355X (gfx950)、ROCm 7.2.3 上从 28 failed 变为 56 passed, 无生产代码、配置或部署改动。

关键文件:

- tests/kernels/moe/test_deep_ep_moe.py (模块 MoE 测试; 类别 test; 类型 test-coverage; 符号 test_deep_ep_moe): 唯一变更文件。在 high-throughput DeepEP

MoE 测试入口通过 monkeypatch 设置 ROCPROFILER_REGISTER_ENABLED=0，把 rocprofiler-sdk 挡在 spawn 出的 worker 之外，绕开 ROCm HSA runtime 在进程退出时的 SIGSEGV。

关键符号：test_deep_ep_moe

关键源码片段

tests/kernels/moe/test_deep_ep_moe.py

唯一变更文件。在 high-throughput DeepEP MoE 测试入口通过 monkeypatch 设置 ROCPROFILER_REGISTER_ENABLED=0，把 rocprofiler-sdk 挡在 spawn 出的 worker 之外，绕开 ROCm HSA runtime 在进程退出时的 SIGSEGV。

```
@pytest.mark.parametrize("dtype", DTYPES)
@pytest.mark.parametrize("m,n,k", MNKS)
@pytest.mark.parametrize("num_experts", [32])
@pytest.mark.parametrize("topk", [6])
@pytest.mark.parametrize("world_dp_size", [(2, 1)])
@pytest.mark.parametrize("per_act_token_quant", [False, True])
@multi_gpu_test(num_gpus=2)
@requires_deep_ep
def test_deep_ep_moe(
    dtype: torch.dtype,
    m: int,
    n: int,
    k: int,
    num_experts: int,
    topk: int,
    world_dp_size: tuple[int, int],
    per_act_token_quant: bool,
    workspace_init,
    monkeypatch, # pytest 内置 fixture：用例结束后自动恢复环境变量，避免影响其他测试
):
    low_latency_mode = False
    use_fp8_dispatch = False

    if current_platform.is_rocm():
        # 仅 ROCm 需要这个 workaround：合作 kernel 启动路径在进程退出时
        # 会在 HSA runtime 中段错误（AqlQueue 析构器写入已 unmap 的信号内存）。
        # torch 隐式挂载 rocprofiler-sdk，spawn 出的子 worker 会继承该注册，
        # 因此这里显式禁用，让 worker 完全不触碰 profiler 路径。
        monkeypatch.setenv("ROCPROFILER_REGISTER_ENABLED", "0")

    set_random_seed(7)
    world_size, dp_size = world_dp_size
    config = TestConfig(dtype=dtype, topk=topk, m=m, k=k, n=n, num_experts=num_experts)

    w1, w2, w1_scale, w2_scale = make_weights(num_experts, n, k, dtype)
```

```
# 通过 multi_gpu_test 生成的多进程 worker 执行 DeepEP MoE kernel 校验
parallel_launch(
    world_size,
    _deep_ep_moe,
    low_latency_mode,
    dp_size,
    config,
    w1,
    w2,
    w1_scale,
    w2_scale,
    use_fp8_dispatch,
    per_act_token_quant,
)
```

评论区精华

核心讨论围绕 workaround 的长期治理：作者 stefankoncarevic 指出根因已在上游修复（ROCm/rocm-systems#6942），并明确提醒在基础镜像带上该修复后，应把本 PR 的环境变量 workaround 与 #51174 的 os._exit workaround 一起移除。合并者 AndreasKaratzas 建议建立 issue 追踪，作者随即创建 #51644 统一记录两个 workaround 的回退条件。

- 根因修复后何时回退 workaround (design): 应合并者要求，作者创建 issue #51644 统一跟踪两个 workaround 的回退条件。
- workaround 的作用范围 (question): reviewer 无异议，AndreasKaratzas 批准合并。

风险与影响

- 风险：
 1. 范围风险低：守卫条件 current_platform.is_rocm() 把影响限制在 ROCm 平台，非 ROCm 工作流完全不受影响。
 2. 维护风险：ROCPROFILER_REGISTER_ENABLED 是 ROCm 运行时内部环境变量，命名与行为可能随版本漂移，workaround 可能失效或变得多余。
 3. 长期依赖风险：该 workaround 依赖上游 ROCm 修复进入基础镜像后的人工回退，存在被遗忘、长期带病运行的可能；issue #51644 用于缓解此风险。
 4. 功能风险：测试进程不再注册 profiler，若未来该测试需要采集 profiler 数据则会被此环境变量阻断，当前测试无此依赖。- 影响：对线上用户无任何影响，改动限定在测试文件。对 ROCm CI 的直接收益是恢复 28 个 DeepEP HT MoE 用例的稳定性，消除与 vLLM 无关的崩溃噪音；对团队而言是一次典型的“外部运行时缺陷 + 临时测试隔离 + 上游跟踪回退”协作模式，后续需按 issue #51644 的条件执行清理。- 风险标记：外部运行时缺陷临时绕过，环境变量隔离仅限测试，上游修复后需主动回退

关联脉络

- PR #51174（上下文未给出具体标题）：PR 评论中与 #51644 一并提及的 os._exit workaround，与本次环境变量 workaround 同根同源，需在上游 ROCm 修复落地后一起回

退。