PR #49242 完整报告

vllm-project/vllm

Stabilize GPU memory teardown between ROCm CI tests

合并时间：2026-07-25 12:21

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49242

执行摘要

- 一句话：修复 ROCm CI 中 GPU 内存残留导致的测试失败
- 推荐动作：建议合并。该 PR 针对明确的 CI 问题进行修复，改动范围限于测试配置和脚本，逻辑清晰，风险可控。

功能与动机

根据 PR 描述（修复 Build 11036 在 mi300 上的失败），失败原因是残存 GPU 内存。具体地：LoRA 测试中 test_qwen3vl_vision_lora 启动时 test_qwen2vl_lora 的 EngineCore 仍持有约 165 GiB 内存（仅 26 GiB 空闲）；Nixl 配置 4（deepseek-vl2-tiny 0.8 util）因前序配置未清理干净导致 prefill 在 cuda:0 上只有 140 GiB 空闲。

实现拆解

1. tests/conftest.py：新增 _should_clean_gpu_memory_between_tests() 函数，当环境变量 VLLM_TEST_CLEAN_GPU_MEMORY 未设置时对 ROCm 平台返回 True（之前默认仅当显式设为 1 才清理），实现默认启用。提取 _wait_for_settled_gpu_memory() 函数，在测试前后分别调用：对 ROCm 使用 wait_for_rocm_memory_to_settle()，对 NVIDIA 保持原 wait_for_gpu_memory_to_clear()。clean_gpu_memory_between_tests fixture 使用这两个辅助函数。
2. tests/lora/test_qwenvl.py：将 Qwen2VLTester 改造为上下文管理器：构造函数中创建 VllmRunner 而非直接 vllm.LLM；__enter__ 返回自身；__exit__ 委托给 _runner.__exit__ 进行清理。所有测试函数（test_qwen2vl_lora、test_qwen2vl_lora_beam_search 等）改用 with Qwen2VLTester(...) as tester: 确保测试结束时 EngineCore 被正常关闭并释放 GPU 内存。
3. tests/v1/kv_connector/nixl_integration/run_accuracy_test.sh：新增 wait_for_gpu_memory_release() 函数，在 ROCm 下调用 Python 的 wait_for_rocm_memory_to_settle()。增强 cleanup_instances()：先 pkill toy_proxy_server，再依次 SIGTERM 和 SIGKILL vllm serve，最后调用内存释放等待。在 run_tests_for_model() 开头也调用一次 cleanup_instances，确保每个模型运行前环境干净。

关键文件：

- tests/conftest.py（模块 测试配置；类别 test；类型 test-coverage；符号 _should_clean_gpu_memory_between_tests, _wait_for_settled_gpu_memory, clean_gpu_memory_between_tests）：核心改动：定义 ROCm 默认清理策略，添加平台感知的内存等待 fixture，是所有测试的入口。
- tests/lora/test_qwenvl.py（模块 Qwen VL 测试；类别 test；类型 test-coverage；符号 _initialize_llm, llm, enter, exit）：LoRA 测试用例改造：通过上下文管理器确保 EngineCore 在测试结束后正确关闭，释放约 165 GiB 显存。
- tests/v1/kv_connector/nixl_integration/run_accuracy_test.sh（模块 Nixl 测试；类别 test；类型 test-coverage）：Nixl 集成测试脚本改进：增加 SIGTERM/SIGKILL 分层终止和 ROCm 内存释放等待，防止跨模型显存污染。

关键符号：_should_clean_gpu_memory_between_tests, _wait_for_settled_gpu_memory, clean_gpu_memory_between_tests, Qwen2VLTester.enter, Qwen2VLTester.exit, wait_for_gpu_memory_release, cleanup_instances

关键源码片段

tests/conftest.py

核心改动：定义 ROCm 默认清理策略，添加平台感知的内存等待 fixture，是所有测试的入口。

```
def _should_clean_gpu_memory_between_tests() -> bool:
    """确定是否应在每个测试前后清理 GPU 内存."""
    setting = os.getenv("VLLM_TEST_CLEAN_GPU_MEMORY")
    if setting == "1":
        return True
    if setting == "0":
        return False
    # ROCm 的 VRAM 回收是惰性的，默认在 ROCm CI 中等待
    return current_platform.is_rocm()

@pytest.fixture(autouse=True)
def clean_gpu_memory_between_tests():
    """自动 fixture：在每个测试前后清理并等待 GPU 内存稳定."""
    if not _should_clean_gpu_memory_between_tests():
        yield
        return

    import gc
    from tests.utils import wait_for_gpu_memory_to_clear, wait_for_rocm_memory_to_settle

    num_gpus = torch.accelerator.device_count()

    def _wait_for_settled_gpu_memory() -> None:
        """等待 GPU 内存回收完成（按平台选择等待机制）."""
        if num_gpus <= 0:
            return
        try:
```

```
            if current_platform.is_rocm():
                wait_for_rocm_memory_to_settle()
            else:
                wait_for_gpu_memory_to_clear(
                    devices=list(range(num_gpus)),
                    threshold_ratio=0.1,
                )
        except ValueError as e:
            logger.info("Failed to clean GPU memory: %s", e)

    # 测试前等待内存稳定
    _wait_for_settled_gpu_memory()

    yield

    # 测试后主动清理并等待
    if torch.cuda.is_available():
        torch.accelerator.empty_cache()
        gc.collect()
    _wait_for_settled_gpu_memory()
```

评论区精华

无实质性讨论。PR 被维护者 AndreasKaratzas 直接批准（LGTM），仅有一个自动化机器人评论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 对 NVIDIA 平台的影响：conftest.py 中 _should_clean_gpu_memory_between_tests 对非 ROCm 平台返回 False（当环境变量未设置时），因此行为不变，但显式区分了平台，可能引入因平台检测错误导致 NVIDIA 也执行等待的风险（概率低）。
  2. 测试总时间增加：自动 fixture 在每次测试前后等待 GPU 内存稳定可能延长执行时间，但可通过设置 VLLM_TEST_CLEAN_GPU_MEMORY=0 跳过来控制。
  3. test_qwenvl.py 的上下文管理器：VllmRunner 的 __exit__ 行为需要保证正确关闭引擎，未覆盖的异常路径可能留下残留进程。
  4. 脚本中的进程终止：pkill -9 可能杀死非目标进程，但当前模式常见，风险可控。 - 影响：目标用户：ROCm 平台下的 CI 测试。效果：减少因残留 GPU 内存导致的 flaky 测试失败，提升 CI 稳定性。侵入性：低，仅影响测试基础设施，不改变 vLLM 核心功能。对 NVIDIA 平台无影响。 - 风险标记：平台特定行为变更，测试总时间可能增加，缺少对异常路径的覆盖

关联脉络

- 暂无明显关联 PR