PR #45490 完整报告

vllm-project/vllm

[ROCm][CI] Make memory sampling less racy in tests and sleep mode

合并时间：2026-06-30 05:26

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45490

执行摘要

- 一句话：修复 ROCm 内存采样竞态条件
- 推荐动作：值得阅读，展示了如何优雅处理平台特定的时序问题，为其他后端类似问题提供参考。

功能与动机

ROCm memory accounting can lag briefly after memory is released, and we were sampling immediately. This leads to flaky test failures like AssertionError: Memory usage increased after sleeping. in sleep mode and premature model loading in hybrid tests (See PR body for job links).

实现拆解

1. GPU Worker sleep 方法(vllm/v1/worker/gpu_worker.py): 在采样内存前后增加 torch.accelerator.synchronize() 确保设备操作完成；将 torch.cuda.mem_get_info 替换为平台无关的 current_platform.mem_get_info；针对 ROCm 增加最多 5 秒的轮询等待循环，每 100ms 检查一次直到内存释放量非负或超时。
2. 测试工具扩展(tests/utils.py): 为 wait_for_gpu_memory_to_clear 新增 stable_duration_s、stable_tolerance_bytes、poll_interval_s 参数，使得在内存降至阈值后继续观察一段时间（默认可选），确保内存释放稳定后再返回。
3. ROCm 专用稳定封装(tests/utils.py): wait_for_rocm_memory_to_settle 调用 wait_for_gpu_memory_to_clear 时启用 stable_duration_s=2.0 和 poll_interval_s=1.0，在混合模型测试中避免因释放滞后导致的错误启动。
4. 其他调整：增加 import time，优化错误消息打印设备信息。

关键文件：

- vllm/v1/worker/gpu_worker.py（模块 GPU 工作器；类别 source；类型 core-logic；符号 sleep）：核心修改：sleep 方法中为 ROCm 添加同步和轮询等待，解决内存采样竞态。
- tests/utils.py（模块 测试工具；类别 test；类型 test-coverage；符号 wait_for_gpu_memory_to_clear, wait_for_rocm_memory_to_settle）：扩展测试工具：为 wait_for_gpu_memory_to_clear 增加稳定窗口参数，并在 wait_for_rocm_memory_to_settle 中启用，防止混合模型测试因内存释放延迟而误判。

关键符号：sleep, wait_for_gpu_memory_to_clear, wait_for_rocm_memory_to_settle

关键源码片段

vllm/v1/worker/gpu_worker.py

核心修改：sleep 方法中为 ROCm 添加同步和轮询等待，解决内存采样竞态。

```
def sleep(self, level: int = 1) -> None:
    # 同步设备，确保之前操作完成
    torch.accelerator.synchronize()
    free_bytes_before_sleep = current_platform.mem_get_info()[0]

    # 保存 level 2 睡眠前的缓冲区
    if level == 2:
        model = self.model_runner.model
        self._sleep_saved_buffers = {
            name: buffer.cpu().clone() for name, buffer in model.named_buffers()
        }

    allocator = get_mem_allocator_instance()
    allocator.sleep(offload_tags=("weights",) if level == 1 else tuple())

    # 再次同步设备
    torch.accelerator.synchronize()
    # 在 ROCm 上，内存记账可能延迟，允许最多等待 5 秒等待释放反映
    deadline = time.monotonic() + (5.0 if current_platform.is_rocm() else 0)
    while True:
        free_bytes_after_sleep, total = current_platform.mem_get_info()
        freed_bytes = free_bytes_after_sleep - free_bytes_before_sleep
        if freed_bytes >= 0 or time.monotonic() >= deadline:
            break
        time.sleep(0.1) # 轮询间隔 100ms

    used_bytes = total - free_bytes_after_sleep
    assert freed_bytes >= 0, "Memory usage increased after sleeping."
    logger.info(
        "Sleep mode freed %s GiB memory, %s GiB memory is still in use.",
        format_gib(freed_bytes),
        format_gib(used_bytes),
    )
```

评论区精华

reviewer tjtanaa 和 mgoin 批准了变更，无额外讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低：引入的轮询等待最多 5 秒（ROCm）且有 deadline 保护，不会死锁；稳定窗口增加测试总耗时（最多额外 2 秒），但超时参数可配置；仅影响 ROCm 路径，CUDA 路径保持不变。
- 影响：正面影响：消除 ROCm CI 中两个已知 flaky 测试（sleep 模式和 hybrid 模型测试），提高持续集成可靠性。轻微负面影响：ROCm 测试可能因等待增加数秒，但可接受。
- 风险标记：ROCM 平台依赖，引入轮询等待，测试时间增加

关联脉络

- PR #47003 [ROCm][CI] Use spawn around the threaded OTLP test: 同为 ROCm CI 稳定性改进，解决线程测试竞态。
- PR #47072 [CI][Bugfix] Add cohere_melody to ROCm test requirements: 同为 ROCm CI 修复，解决依赖缺失问题。
- PR #46990 [ROCm][DeepEP] Stabilize high-throughput DBO for DP+EP: 同为 ROCm 稳定性修复，解决 DBO 精度问题。