

PR #40686 完整报告

vllm-project/vllm

fix(rocm): remove workaround causing invalid argument on Qwen3.5 with TP=2

合并时间: 2026-05-06 16:38

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/40686>

执行摘要

该 PR 移除了 ROCm 平台上 `csrc/cumem_allocator.cpp` 中一段虚拟地址循环 workaround, 修复了 Qwen3.5 模型在 `tensor_parallel_size=2` 且启用 sleep mode 时 `wake_up` 报 `invalid argument` 的错误。改动仅删除 22 行代码, 不涉及新增逻辑。

功能与动机

在 ROCm 平台上, 使用 Qwen3.5 模型并设置 `enable_sleep_mode=True`, `tensor_parallel_size=2` 时, 调用 `llm.wake_up()` 会触发 `invalid argument` 错误。作者提供了可复现脚本, 问题根源是 `csrc/cumem_allocator.cpp` 中一段专门用于 ROCm 的 workaround 代码, 该代码通过 `cuMemAddressFree` → `cuMemAddressReserve` 循环尝试强制释放物理 VRAM, 但在多卡场景下导致虚拟地址分配竞争, 从而返回错误。

实现拆解

1. 定位问题代码: 在 `csrc/cumem_allocator.cpp` 文件的 `unmap_and_release` 函数中, ROCm 分支末尾有一段约 22 行的 workaround (`#else` 块内部), 注释明确说明其目的是通过 "释放虚拟地址→立即重新预留同一地址" 的方式来迫使 ROCm 驱动归还物理内存。
2. 移除 workaround: 直接删除该代码块, 保留正常的错误处理逻辑 (若 `first_error != no_error` 则调用 `CUDA_CHECK`)。该 workaround 在较新的 ROCm 驱动版本中已不再需要, 反而因多进程 / 多线程竞争导致 `invalid argument`。
3. 验证与合并: 仓库维护者 `tjtanaa` 审核后批准, 未要求新增测试或其他更改。PR 包含两次 Merge branch 'main' 提交, 保持与主线同步。

`csrc/cumem_allocator.cpp`

本次 PR 唯一修改的文件, 删除了 ROCm 平台下 `unmap_and_release` 函数末尾的一段 workaround 代码 (22 行), 该代码导致 Qwen3.5 在 TP=2 时 `wake_up` 失败。

```
// csrc/cumem_allocator.cpp
```

```
void unmap_and_release(unsigned long long device, ssize_t size,
                        unsigned long long d_mem, void** p_memHandle,
                        size_t* chunk_sizes, int num_chunks) {
```

```
// ... (前面的 unmap 和 release 循环保持不变)
```

```
for (auto i = 0; i < num_chunks; ++i) {
    CUresult status = cuMemRelease(*(p_memHandle[i]));
    if (status != no_error && first_error == no_error) {
```

```

    first_error = status;
}
}

// 以下为本次 PR 删除的内容（原 workaround）：
// // ROCm workaround: hipMemRelease does not return physical VRAM to the
// // free pool while the virtual-address reservation is still held.
// // Cycling cuMemAddressFree → cuMemAddressReserve (at the same address)
// // forces the driver to actually release the physical pages while keeping
// // the same VA available for a later create_and_map.
// if (first_error == no_error) {
//     first_error = cuMemAddressFree(d_mem, size);
//     if (first_error == no_error) {
//         CUdeviceptr d_mem_new = 0;
//         first_error = cuMemAddressReserve(&d_mem_new, size, 0, d_mem, 0);
//         if (first_error == no_error && d_mem_new != d_mem) {
//             cuMemAddressFree(d_mem_new, size);
//             snprintf(error_msg, sizeof(error_msg),
//                  "ROCm: VA re-reserve got %p instead of %p", (void*)d_mem_new,
//                  (void*)d_mem);
//             error_code = CUresult(1);
//             std::cerr << error_msg << std::endl;
//             return;
//         }
//     }
// }

// 删除 workaround 后，直接检查错误并传播
if (first_error != no_error) {
    CUDA_CHECK(first_error);
}
}

```

评论区精华

该 PR 的 review 讨论较少，仅有的两位 bot 评论（Claude 和 Gemini）均未提供实质反馈。仓库维护者 [tjtanaa](#) 直接批准 ("LGTM")。未发现关于 workaround 移除后内存管理正确性或潜在回归的深入讨论。

风险与影响

- 回归风险（中等）：该 workaround 原本用于解决旧版本 ROCm 驱动中物理内存不释放的问题。移除后，在较旧的 ROCm 驱动版本上可能出现内存泄漏。
- 测试覆盖不足：PR 未增加任何测试用例，对于核心内存管理器的更改来说，缺少回归测试是一个风险。
- 影响范围：仅影响 ROCm 平台中启用 sleep mode 的场景，CUDA 路径完全不受影响。

关联脉络

近期有多项针对 ROCm 平台的修复和改进（如 PR#41387、PR#40575、PR#40871），表明团队正在持续提升 ROCm 支持的稳定性。本 PR 与这些变更共同完善了 ROCm 平台的基础设施支持，特别是 sleep/wake_up 循环功能的可靠性。