PR #46958 完整报告

vllm-project/vllm

[BugFix] Revert "[KV Offload] Use background thread for mmap / cpu_tensors pinning"

合并时间：2026-06-29 22:54

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46958

执行摘要

- 一句话：回退线程式 mmap pinning 以修复 CUDA Graph 挂起
- 推荐动作：该 PR 值得仔细阅读，因为它揭示了 CUDA Graph capture 与内存操作的严格同步要求。设计决策上，在正确性与微性能优化之间选择了正确性。建议关注后续是否会有更完善的异步方案加入。

功能与动机

Issue #46933 报告启用 KV offload 和 CUDA Graph 后，在 capture 阶段发生永久挂起。根本原因是 #45850 将 cudaHostRegister 操作移至后台线程，而 CUDA Graph capture 要求内存 pinning 状态在 capture 前完全确定，后台线程的延迟执行导致竞争条件，引发死锁。

实现拆解

实现拆解如下：

1. 删除后台线程相关的导入和成员：在 gpu_worker.py 中移除 import threading，从 SingleDirectionOffloadingHandler.__init__的参数列表中删除 pin_thread 和 manually_pinned_tensors，并删除对应的成员变量。
2. 移除 _pin_cpu_tensors 函数：删除该函数（#45850 新增），该函数负责在后台线程中注册 CPU tensor 的 pinned memory。
3. 恢复同步的 pin_mmap_region：重新引入一个简洁的 pin_mmap_region 函数，在调用线程（即 GPUWorker 初始化线程）中直接调用 cudaHostRegister，并设置 region.is_pinned = True。该函数包含对非 CUDA 平台的跳过逻辑。
4. 简化 OffloadingWorker.init：移除后台线程的创建和启动逻辑，改为在初始化 mmap_region 时直接调用 pin_mmap_region(mmap_region)（如果 pin_memory 为 True）。同时移除成员变量 self.pin_thread 和 self._manually_pinned_tensors。
5. 清理 shutdown 方法：移除 shutdown 中对线程 join 以及循环调用 cudaHostUnregister 的逻辑，简化关闭流程。

关键文件：

- vllm/v1/kv_offload/cpu/gpu_worker.py（模块 KV 卸载；类别 source；类型 core-logic；符号 pin_mmap_region, _pin_cpu_tensors）：单个变更文件，包含核心 KV offload 调度逻辑的调整。删除了后台线程 pinning，恢复为同步 pinning，直接影响 KV offload 与

CUDA Graph 的兼容性。

关键符号：pin_mmap_region, _pin_cpu_tensors, SingleDirectionOffloadingHandler.init, SingleDirectionOffloadingHandler.shutdown, OffloadingWorker.init

关键源码片段

vllm/v1/kv_offload/cpu/gpu_worker.py

单个变更文件，包含核心 KV offload 调度逻辑的调整。删除了后台线程 pinning，恢复为同步 pinning，直接影响 KV offload 与 CUDA Graph 的兼容性。

```
# SPDX-License-Identifier: Apache-2.0

# pin_mmap_region 函数 —— 同步注册 mmap 内存为 CUDA 固定页
def pin_mmap_region(region: SharedOffloadRegion) -> None:
    # Register the entire mmap as CUDA pinned memory via cudaHostRegister.
    # 如果不是 CUDA / ROCm 平台则跳过
    if not current_platform.is_cuda_alike():
        logger.info(
            'Skipping mmap host registration on %s; cudaHostRegister is only '
            'available on CUDA/ROCm.',
            current_platform.device_name,
        )
        return

    rank = region.rank
    base_ptr = region._base.data_ptr()
    # 同步调用 cudaHostRegister 注册整块 mmap 内存
    result = torch.cuda.cudart().cudaHostRegister(base_ptr, region.total_size_bytes, 0)
    if result.value != 0:
        logger.warning(
            'cudaHostRegister failed for rank=%d (code=%d) — '
            'transfers will still work but may be slower (unpinned DMA)',
            rank,
            result,
        )
    else:
        logger.debug(
            'cudaHostRegister rank=%d %.2f GB',
            rank,
            region.total_size_bytes / 1e9,
        )
        region.is_pinned = True # 标记为已固定，后续不再尝试


# SingleDirectionOffloadingHandler 构造函数 —— 移除线程相关参数
def __init__(
    self,
    gpu_tensors: list[torch.Tensor],
```

```
    cpu_tensors: list[torch.Tensor],
    block_size_factor: int,
    kv_cache_groups_data_refs: list[list[CanonicalKVCacheRef]],
    gpu_to_cpu: bool,
    mmap_region: SharedOffloadRegion | None = None,
):
    # ... ( 初始化逻辑不变，但删除了 self._pin_thread 和 self._manually_pinned_tensors)
    self._mmap_region = mmap_region
    # 原 self._pin_thread.join() 等代码已删除
    # ...


# shutdown 方法 —— 移除线程 join 和手动 unpin 的循环
def shutdown(self) -> None:
    # ...
    self._buffer_pool.clear()
    # 原 self._pin_thread.join() 等代码已删除
    self.src_tensors.clear()
    self.dst_tensors.clear()
    if self._mmap_region is not None:
        self._mmap_region.cleanup()
        self._mmap_region = None
    # ...


# OffloadingWorker.__init__ —— 同步执行 pinming
def __init__(self, mmap_region: SharedOffloadRegion | None = None):
    pin_memory = PIN_MEMORY
    # 移除了 self.pin_thread 等成员
    if mmap_region is not None and pin_memory:
        # 直接在初始化线程中调用 pin_mmap_region ，不再创建后台线程
        pin_mmap_region(mmap_region)
    # 继续其他初始化 ...
```

评论区精华

主要讨论：

- CI 失败：作者 varun-sundar-rabindranath 在 issue 评论中指出，一个 CI 测试失败（buildkite/ci/pr/async-engine-inputs-utils-worker-config-cpu）在 main 分支上也同样失败，请求强制合并。该请求被 @orozery 批准。
- 无其他讨论：PR 没有收到其他 review 评论，整体改动清晰（纯 revert），社区一致同意快速修复。
- CI 测试失败分析 (other): 维护者 orozery 批准 PR，认为 CI 失败与本次变更无关。

风险与影响

- 风险：技术风险：

1. 回归风险：回退 #45850 后，mmap pinning 操作从后台线程改为同步执行，可能略微增加 GPU 初始化延迟。但此延迟通常可接受，且恢复了与 CUDA Graph 的兼容性。
2. 缺少测试覆盖：PR 没有新增测试验证 CUDA Graph + KV offload 正常工作。建议后续添加 regression test。
3. 平台兼容性：pin_mmap_region 包含对 is_cuda_alike 的判断，非 CUDA 平台跳过，无影响。 - 影响：影响范围：

- 用户：使用 KV offload + CUDA Graph 的用户将不再遇到 capture 挂起问题。不使用 CUDA Graph 或 KV offload 的用户无感知。
- 系统：初始化期间的内存 pinning 变为同步，可能影响服务启动时间，但通常不明显。
- 团队：明确了一个重要的设计约束：CUDA Graph capture 期间不能有异步内存操作。后续若重新引入异步 pinning 需要额外同步机制。
- 风险标记：回归风险，缺少测试覆盖，核心路径变更

关联脉络

- PR #45850 [KV Offload] Use background thread for mmap / cpu_tensors pinning: 此 PR 回退了 #45850 的改动，是其直接反操作。
- PR #46933 [Bug]: CPU KV-Offloading CUDA Graph capture hang: 该 issue 报告了启用 #45850 后 CUDA Graph capture 挂起的问题，是此修复的触发来源。