# PR #35882 完整报告

sgl-project/sglang

[diffusion] optimization: transfer mapped layers through a courier thread

合并时间：2026-08-22 17:17

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35882

## 执行摘要

- 一句话：映射层权重经 courier 线程异步搬运，去噪步骤提速约 31%
- 推荐动作：值得精读。核心看点：生产者 - 消费者双缓冲 courier 的设计（event 驱动 slot 复用、异常经结果通道发布、worker 只备张量不动模块状态、kill switch + 故障回退 + 阻塞直通三层兜底），以及"用测试环境变量把 CI 大卡压成消费级预算"的测试方法论。该模式可复用到其他 offload/ 预取场景，如后续的 VAE 常驻与主机内存预算守卫。

## 功能与动机

PR body 明确指出问题本质："A copy whose source is a checkpoint mapping is synchronous however it is requested — the driver stages unpinned memory through its own buffer — and the prefetch hooks run on the compute thread, so every mapped layer stalled the denoise step for its whole transfer." 在主机内存预算不足时，LayerwiseOffloadManager 会把部分层权重留在 checkpoint 文件映射上（_mapped_cpu_weights，由页缓存决定驻留），这类权重未 pin，往 GPU 的拷贝先经驱动 staging buffer，因此无论 non_blocking 如何设置都是同步的；而 prefetch hook 又跑在计算线程上，导致每个映射层都让 denoise 步骤等待整次传输。PR 目标是把这类传输挪到工作线程与计算重叠，并证明输出等价（视频 SSIM 0.9941 与基线自身 run-to-run 一致、音频流逐位相同）。

## 实现拆解

1. 状态与入口改造：LayerwiseOffloadManager.__init__（layerwise_offload.py）新增 _mapped_courier 与 _courier_inflight 两个状态字段；prefetch_layer() 顶部增加短路分支——层已在 courier 中时，非阻塞调用直接返回，阻塞调用转入 _collect_mapped_layer() 等待并绑定参数。
2. 核心机制：MappedLayerCourier 类：新增工作线程类。构造函数按所有映射层的最大字节数分配 2 个 pin_memory slot 组成双缓冲，并启动 daemon 线程 _run() 消费任务队列；submit() 幂等入队，_ship() 在复用 slot 前先对上一个 event 执行 synchronize()，再把各张量按 dtype 对齐写入 slot 视图（页缓存命中时即普通 memcpy），随后在独立 stream 上逐个发起 non_blocking=True 的设备拷贝并 event.record()；collect() 用 threading.Condition 阻塞等待结果，异常通过 _results 发布（published, never swallowed）而非吞掉。设计关键：worker 线程只准备设备张量与 event，参数绑定（target.data = ...）只发生在计算线程的 collect 阶段，模块状态永不被并发触碰。

3. 接入 prefetch_layer 与回退路径：非阻塞请求且该层含映射权重时，_ensure_mapped_courier() 创建 courier（内部读取 kill switch）并 submit()，标记 ship_mapped 后映射权重在直接拷贝循环中被跳过；阻塞调用方始终走原直接同步路径。SGLANG_DIFFUSION_DISABLE_MAPPED_COURIER（envs.py 新增，_lazy_bool）可强制禁用 courier；release_all() 会 drain in-flight 层并关闭 courier，避免设备张量残留在 _results 中；broken courier 会被撤换并回退同步拷贝。
4. CI 复现消费级约束的测试配套：新增两个测试专用环境变量——SGLANG_DIFFUSION_TEST_FORCE_HOST_AVAILABLE_GIB 让 host_memory_budget.py 的 host_memory_available_bytes() 按"伪装机 GiB − 进程匿名内存（/proc/self/status 的 RssAnon）"计算可用量，SGLANG_DIFFUSION_TEST_CAP_DEVICE_MEMORY_GIB 在 gpu_worker.py 的 _cap_device_memory_for_tests() 中通过 set_per_process_memory_fraction 把缓存分配器压到目标显存。
5. 测试与性能基线：test_layerwise_offload.py 新增 4 个单测覆盖 ship/collect 握手、kill switch、release_all drain、courier 中途死亡回退；gpu_cases.py 新增 minimax_h3_t2va_consumer_budget_1gpu_5090 e2e 用例（伪 12 GiB 显存 / 32 GiB 主机、video_vae=24 常驻）；5090.json 写入该用例的 stage/denoise/e2e 时间与峰值显存基线（12288 MB，任何超预算变更直接失败）。

关键文件：

- python/sglang/multimodal_gen/runtime/managers/memory_managers/layerwise_offload.py（模块 层间卸载；类别 source；类型 core-logic；符号 MappedLayerCourier, MappedLayerCourier.submit, MappedLayerCourier.collect, MappedLayerCourier._ship）：核心变更文件：新增 MappedLayerCourier 工作线程类（双 pinned slot、独立 stream、event 同步、异常发布），并改造 prefetch_layer / release_all 接入 courier，是本 PR 性能收益与正确性保证的源头。
- python/sglang/multimodal_gen/envs.py（模块 环境变量；类别 source；类型 configuration；符号 _lazy_optional_float, SGLANG_DIFFUSION_DISABLE_MAPPED_COURIER, SGLANG_DIFFUSION_TEST_FORCE_HOST_AVAILABLE_GIB, SGLANG_DIFFUSION_TEST_CAP_DEVICE_MEMORY_GIB）：新增 3 个环境变量：kill switch（SGLANG_DIFFUSION_DISABLE_MAPPED_COURIER）与两个测试钩子（SGLANG_DIFFUSION_TEST_FORCE_HOST_AVAILABLE_GIB / SGLANG_DIFFUSION_TEST_CAP_DEVICE_MEMORY_GIB），并新增 _lazy_optional_float 惰性解析辅助；是回退开关与 CI 约束的配置入口。
- python/sglang/multimodal_gen/test/unit/test_layerwise_offload.py（模块 层间卸载；类别 test；类型 test-coverage；符号 test_mapped_layers_ship_through_the_courier, test_the_courier_kill_switch_forces_the_synchronous_path, test_release_all_drains_the_courier, test_a_broken_courier_falls_back_to_the_synchronous_copy）：4 个新单测精确覆盖 courier 的关键正确性契约：异步提交不阻塞、阻塞 collect 完成绑定、kill switch 强制同步、release_all drain、broken courier 回退同步拷贝；前两者在禁用 courier 时必然失败，是行为契约的守护。

- python/sglang/multimodal_gen/runtime/managers/gpu_worker.py（模块 显存管理；类别 source；类型 core-logic；符号 _cap_device_memory_for_tests, init_device_and_model）：新增 _cap_device_memory_for_tests：通过 set_per_process_memory_fraction 把 CI 大卡压到用例目标显存，让“峰值 VRAM 基线 = 预算”从推断变为强制执行；这是 12 GiB 预算用例能成立的硬件前提。
- python/sglang/multimodal_gen/runtime/managers/memory_managers/host_memory_budget.py（模块 主机内存；类别 source；类型 configuration；符号 host_memory_available_bytes）：host_memory_available_bytes() 增加测试强制分支：按“伪装机 GiB − 进程 RssAnon”计算可用内存，让 CI runner 真实的充裕内存不再绕过受限放置路径（mapped 权重、部分 pin、courier）。
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 服务端测试；类别 test；类型 test-coverage；符号 _make_5090_h3_consumer_budget_case）：新增 minimax_h3_t2va_consumer_budget_1gpu_5090 e2e 用例，一次性看护整个受限放置栈：VAE 留 mapping、小预算逐层 pin、courier 传输、decode 期间 VAE 常驻。
- python/sglang/multimodal_gen/test/unit/test_host_memory_budget.py（模块 主机内存；类别 test；类型 test-coverage；符号 test_the_forced_host_size_behaves_like_a_machine_of_that_size）：验证 FORCE_HOST_AVAILABLE_GIB 分支的语义：可用内存严格落在伪装机大小内，且机器翻倍则可用量增加一个机器。
- python/sglang/multimodal_gen/test/server/perf_baselines/5090.json（模块 性能基线；类别 test；类型 test-coverage）：新用例的 stage / denoise / e2e 时间与峰值显存基线；峰值显存基线设为预算本身 12288 MB，任何超预算变更直接失败。

关键符号：MappedLayerCourier, MappedLayerCourier.submit, MappedLayerCourier.pending, MappedLayerCourier.collect, MappedLayerCourier.close, MappedLayerCourier._run, MappedLayerCourier._ship, LayerwiseOffloadManager.prefetch_layer, LayerwiseOffloadManager._ensure_mapped_courier, LayerwiseOffloadManager._collect_mapped_layer, LayerwiseOffloadManager.release_all, gpu_worker._cap_device_memory_for_tests, host_memory_budget.host_memory_available_bytes, envs._lazy_optional_float, gpu_cases._make_5090_h3_consumer_budget_case

## 评论区精华

该 PR 没有 review 评论与 issue 讨论（comments_count = 0、review_comments_count = 0），唯一可见的协作信息来自 PR body 的 merge note：

> "Merge note: this PR and #35867 touch the same file; whichever lands second rebases (the resolution is mechanical — both sides add independent methods)."

即本 PR 与 #35867（per-layer pinning）都改动 layerwise_offload.py，但各自只新增独立方法，后合入者 rebase 即可。PR body 还记录了关键设计结论：阻塞调用方与 courier 失败都保留直接同步路径；worker 线程绝不触碰模块状态；输出等价通过视频级 SSIM 与音频逐位比对验证。

- 与 #35867 的同文件合并冲突处理 (other): 先合入者直接通过，后合入者 rebase 即可机械解决冲突；两侧无逻辑交叠。

## 风险与影响

- 风险：
  - 线程并发与 fork 兼容性：diffusion worker 默认 multiprocess 方法为 fork（envs.py 中 SGLANG_DIFFUSION_WORKER_MULTIPROC_METHOD 默认 "fork"），courier 是每 manager 懒创建的 daemon 线程；若未来出现“创建 courier 后再 fork”的路径，子进程会丢失线程且 Condition 状态可能残留。当前创建时机在 worker 初始化后，风险可控但需要留意。
  - 槽位复用依赖 event.synchronize()：_ship() 假设设备具备 CUDA-like stream/event 语义，在 MPS 或非 CUDA 后端上该机制未被证明。
  - 性能依赖页缓存命中：courier 快的前提是映射页仍在页缓存中；冷启动或主机内存压力大时 window.copy_() 会退化为读盘，可能不如直接同步路径。
  - 测试钩子进入生产路径：SGLANG_DIFFUSION_TEST_FORCE_HOST_AVAILABLE_GIB 在 host_memory_available_bytes() 中优先于 cgroup 逻辑，SGLANG_DIFFUSION_TEST_CAP_DEVICE_MEMORY_GIB 会调用 set_per_process_memory_fraction；生产环境误设会直接改变内存规划或限制显存。
  - 性能基线波动：5090.json 中 denoise 基线按两次 CI 运行 69–101 s 的观测放宽，perf 用例存在 flaky 风险；但峰值显存基线设为预算本身（12288 MB），超预算即失败，是强约束。
  - 新增 pinned 内存压力：2 个 slot 各为最大映射层大小，pinned 内存不可换页，会增加主机内存压力。
  - 影响：用户侧：启用 --dit-cpu-offload 且权重保留在 checkpoint mapping 的场景（典型如 MiniMax-H3 消费级 4090/5090 内存预算），denoise 中位数从 25 s/step 降到 18 s/step（约 31%）；与 #35867、#35862 叠加后同预算可跑 lossless bf16，11.0–12.0 s/step 超过 ComfyUI 的 13.0 s/step。系统侧：新增一个常驻 daemon 线程与最多 2 × 最大映射层字节的 pinned 内存，release_all 保证 in-flight 层被 drain。团队侧：新增 5090 消费级预算 e2e 性能看护用例与两个 test-only 环境变量，CI 可强制复现消费级约束；影响范围集中在 sglang-diffusion 的 layerwise offload 路径，不触碰 SRT 主链路。
  - 风险标记：线程并发模型，核心路径变更，测试钩子进入生产路径，性能基线波动，依赖页缓存命中

## 关联脉络

- PR #35867 per-layer pinning: PR body 明确声明与之叠加（Stacked with per-layer pinning），且与本 PR 同改 layerwise_offload.py，存在 merge note 说明的机械可解冲突。
- PR #35862 VAE staying on its mapping: PR body 声明与之叠加；三者组合后同一预算可跑 lossless bf16 达到 11.0–12.0 s/step。

- PR #36051 [diffusion] CI: guard the anonymous-host budget alongside peak VRAM: 同为 5090.json 与服务端测试集的 CI 预算守卫工作，延续本 PR 引入的“约束复现”测试思路。
- PR #36034 [Diffusion] UX: clean up startup and offload logs: 后续对同模块 layerwise_offload.py、host_memory_budget.py 的清理与重构，说明 offload 栈持续迭代。