PR #50094 完整报告

vllm-project/vllm

[KV Offload] Move CPUOffloadingSpec onto SharedOffloadRegion

合并时间：2026-07-29 18:05

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/50094

执行摘要

- 一句话：CPUOffloadingSpec worker 改用 SharedOffloadRegion mmap 替代私有 tensor
- 推荐动作：值得精读，尤其是 create_worker 的改造思路——用共享 mmap 替换私有 pinned tensor 在大型部署中很典型，也展示了为未来功能预留扩展点的设计方法。

功能与动机

解决 PyTorch CUDACachingHostAllocator 在大型固定分配时按 2 的幂取整导致的内存过度消耗（参见 #48436 诊断），并为 TP 去重（#47929）提供共享内存区域基础设施。PR body 引用 @orozery 的授权，确认本 PR 可独立合并。

实现拆解

1. 对齐调整：将 CPUOffloadingSpec.BLOCK_SIZE_ALIGNMENT 从 1 改为 SharedOffloadRegion.BLOCK_SIZE_ALIGNMENT，使块编号与 mmap 页面大小对齐，确保调度器和 worker 计数一致。
2. worker 创建逻辑：修改 create_worker 方法，在 CUDA/ROCm 平台且 num_blocks > 0 时，创建 SharedOffloadRegion 实例，将物理设备索引折叠到 [0, world_size) 槽位，并传递给 CPUOffloadingWorker。空缓存或非 CUDA 平台保持私有 tensor 路径。
3. worker 构造：CPUOffloadingWorker 现在接收可选的 mmap_region 参数，当为 None 时使用原有张量分配。
4. 测试覆盖：修改现有测试以反映新的对齐值，并新增三个测试验证 mmap 路径、非 CUDA 回退和空缓存行为。 无其他配置或部署配套变更。

关键文件：

- vllm/v1/kv_offload/cpu/spec.py（模块 KV 卸载；类别 source；类型 core-logic；符号 CPUOffloadingSpec, create_worker）：核心变更文件，修改了 CPUOffloadingSpec 的 worker 创建逻辑和对齐方式
- tests/v1/kv_offload/test_factory.py（模块 卸载测试；类别 test；类型 test-coverage；符号 test_cpu_spec_create_worker_uses_mmap_on_cuda_alike, test_cpu_spec_create_worker_uses_tensor_path_off_cuda_alike, test_cpu_spec_create_worker_skips_mmap_for_empty_cache, fake_region_ctor）：测试配套，新增三个测试用例覆盖 mmap 路径、非 CUDA 回退和空缓存场景，并调整现有测试适应新的对齐

关键符号：CPUOffloadingSpec.create_worker

关键源码片段

vllm/v1/kv_offload/cpu/spec.py

核心变更文件，修改了 CPUOffloadingSpec 的 worker 创建逻辑和对齐方式

```
# vllm/v1/kv_offload/cpu/spec.py

class CPUOffloadingSpec(OffloadingSpec):
    # 从 1 改为 mmap 页面大小，保证块编号与 mmap 对齐
    BLOCK_SIZE_ALIGNMENT = SharedOffloadRegion.BLOCK_SIZE_ALIGNMENT
    SUPPORTS_REPLICATED_LAYOUT = False

    def create_worker(self, kv_caches: CanonicalKVCaches) -> CPUOffloadingWorker:
        mmap_region: SharedOffloadRegion | None = None
        # 空缓存（num_blocks==0）无法 mmap 零字节，回退到 tensor 路径
        if current_platform.is_cuda_alike() and self.num_blocks > 0:
            # 使用共享 mmap 区域替代 per-rank 私有 pinned tensor
            world_size = self.config.parallel.world_size
            rank = torch.accelerator.current_device_index() % world_size
            mmap_region = SharedOffloadRegion(
                engine_id=self.config.engine_id,
                num_blocks=self.num_blocks,
                rank=rank,
                kv_bytes_per_block=self.kv_bytes_per_chunk,
                cpu_page_size=self.cpu_page_size_per_worker,
            )
        return CPUOffloadingWorker(
            kv_caches=kv_caches,
            blocks_per_chunk=self.blocks_per_chunk,
            num_cpu_blocks=self.num_blocks,
            mmap_region=mmap_region, # 非 CUDA 或空缓存时为 None，保持旧路径
        )
```

评论区精华

PR 无实质性的 review 讨论。@orozery 批准了变更，并在关联 issue 中确认此独立 PR 的计划（issue comment #5096162285）。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 空缓存回退：num_blocks == 0 时 mmap 无法创建零字节区域，若回退路径存在 bug 可能导致进程崩溃。

2. 对齐兼容性：新对齐可能使 num_blocks 轻微降低，需确保调度器与 worker 双方对齐一致。
3. 平台差异：非 CUDA/ROCm 平台（如 XPU）保持旧路径，若后续引入 CUDA 专属 bug 不影响它们。
4. 共享内存竞争：SharedOffloadRegion 使用 /dev/shm，多进程环境下若清理不当可能导致内存泄漏或状态不一致。 - 影响：用户视角：内存使用更可预测，不再因 power-of-2 取整导致意外高 RSS；系统视角：默认 KV offload 后端内部重构，为后续 TP 去重功能（#47929）铺平道路；团队视角：减少对 PyTorch 分配器的依赖，提升可控性。XPU 用户无影响。 - 风险标记：空缓存回退路径，对齐兼容性，平台差异路径，共享内存竞争条件

关联脉络

- PR #47929 [Feature]: Deduplicate replicated MLA KV across TP ranks in native offloading: 本 PR 是 TP 去重功能的基础分配步骤，为 #47929 所需的共享内存区域做准备
- PR #48436 [KV Offload] Bypass power-of-2 rounding in KV offload CPU pinned allocation: 本 PR 从根源上解决了 #48436 诊断的 CUDACachingHostAllocator 膨胀问题，并 supersede 了 #48436 的临时 fix