

PR #56845 完整报告

vllm-project/vllm

[Refactor] Remove dead kernel code

合并时间: 2026-09-16 04:17

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56845>

执行摘要

- 一句话: 清理 8 文件 1895 行死内核代码, 覆盖 AMD 与量化算子路径
- 推荐动作: 值得快速浏览, 不必精读。它展示了 vLLM 中临时代码生命周期的管理实践: 在 docstring 中写明删除条件 (如 AITER 修复版本), 到期后由维护者执行清理。对于内核与算子维护者, 建议关注两个设计决策: 一是 lamport.py 中基于 PDL 的死锁规避证明方法, 这是跨 GPU 通信 kernel 的高价值设计文档, 即使代码被删除也值得从 git 历史中学习; 二是 triton_fp8_mqa_logits.py 中针对 gfx942 LDS 预算的 tile 大小自适应判断, 展示了 Triton kernel 在特定硬件上做 occupancy 与硬件上限双重校验的思路。另外, 本 PR 完全没有 review 讨论, 阅读时建议自行核对删除边界, 或在需要修改这些目录时通过 git blame 追溯删除决策。

功能与动机

PR body 仅一句话 "Purpose Remove dead kernel code", 未关联 Issue。从删除内容看, 这些代码均已完成历史使命: triton_fp8_mqa_logits.py 的 docstring 明确写明是 "Temporary gfx942 fallback for AITER's fp8_mqa_logits kernel", 并留下 "TODO: Remove this vendored copy once vLLM pins an AITER version that includes ROCm/aiter#3257 bugfix" 的到期删除约定; turboquant_decode_v2 中被删除的 stage1 v2 实现属于被 unified 路径取代的旧方案; Lamport 融合内核与 mm_towers 算子则是因为对应模型路径已不再调用。删除这些代码是清理技术债、降低后续维护与检索成本的自然动作。

实现拆解

变更按文件粒度分为整文件删除与局部代码块删除两类, 整体思路是先确认目标代码已无引用, 再执行删除并靠 CI 验证。

1. 整文件删除 (3 个文件, 共 1218 行):

- vllm/models/inkling/amd/ops/lamport.py (-766 行): 整体移除包含 _publish_input_kernel、_reduce_insert_kernel、_sconv_publish_kernel、_gather_norm_kernel、_validate_lamport_init_kernel 等 8 个 Triton kernel 的融合内核模块。该模块实现基于 PDL (Programmatic Dependent Launch) 的死锁规避跨卡通信, docstring 详尽记录了 liveness 论证, 是本次删除中设计最复杂的部分。
- vllm/models/inkling/amd/ops/mm_towers.py (-190 行): 移除 dmel_embed_sum_norm 与 rmsnorm_gelu 两个融合算子及其 Triton kernel, 这些算子

曾是 Inkling 视觉 / 音频 tower 的提速路径。

- `vllm/v1/attention/ops/triton_fp8_mqa_logits.py` (-262 行) : 删除 vendored 的 AITER `fp8_mqa_logits gfx942` 回退实现。该文件在 docstring 中已预告删除条件 (vLLM 固定的 AITER 版本包含 ROCm/aiter#3257 修复), 本次删除意味着该条件已满足。

2. 局部代码块删除 (5 个文件, 共 677 行) :

- `vllm/v1/attention/ops/turboquant_soa/triton_turboquant_decode_v2.py` (-572 行) : 删除 `_tq_decode_stage1_v2` 这一个庞大的 Triton kernel 及其相关常量与导入清理。该 kernel 是 stage1 v2 方案的完整实现 (含 FP8/MSE 反量化、pair LUT、SoA 布局等复杂逻辑), 被删除后文件仍保留 v1 路径。
- `vllm/models/glm5next/amd/ops/kpool_compress.py` (-44 行) : 删除 `compute_pooled_write_locs` 与 `build_pooled_page_table` 两个辅助函数, 它们用于 pool id 到物理 slot 的映射, 说明对应的调用方已被替换或移除。
- `vllm/models/kimi_k3/nvidia/ops/cute_dsl/latent_moe_tail/primitives.py` (-31 行) : 删除 `block_sum_specialized`, 这是与上游等价的 FP32 block reduction 原语, 说明 K3 的 MoE tail 不再需要该归约路径。
- `vllm/model_executor/layers/mamba/ops/replayssm_config.py` (-15 行) : 删除 `override_replayssm_config contextmanager`, 该工具用于在测试中临时固定 kernel launch 配置, 删除后 `_overrides` 字典仍在但不再暴露外部修改入口。
- `vllm/v1/attention/ops/turboquant_soa/triton_turboquant_unified_attention.py` (-15 行) : 删除 `_get_pit_in_query_dtype` 及其模块级 `_pit_qdtype_cache` 缓存字典, 移除按 `data_ptr` 缓存 PiT 类型转换的路径。

3. 验证与合并: 作者通过两次 `/ci run` 触发 Buildkite CI (#88844 与 #89121), 第二次提交为 `mergify[bot]` 的 main 分支合并, 最终由 khluu 合并、sfeng33 审批通过。测试配套方面没有新增或修改的测试文件, 说明本次删除依赖的是既有路径的编译与运行回归覆盖。

关键文件:

- `vllm/models/inkling/amd/ops/lamport.py` (模块 AMD 算子; 类别 infra; 类型 deletion; 符号 `_pack_bf16_pairs`, `_unpack_bf16_pairs`, `_wait_pairs`, `_publish_input_kernel`) : 整文件删除 766 行, 是本次清理中规模最大、设计最复杂的部分。该模块是基于 PDL 的死锁规避融合内核 (RS + short-conv + AG + residual + RMSNorm), 包含 8 个 Triton kernel 与完整的 liveness 论证文档, 删除它标志着 Inkling AMD 的 Lamport 路径正式退役。
- `vllm/v1/attention/ops/triton_fp8_mqa_logits.py` (模块 MQA 算子; 类别 infra; 类型 deletion; 符号 `_gfx942_default_tile_fits_lds`, `_fp8_mqa_logits_kernel`, `fp8_mqa_logits_gfx942`) : 整文件删除 262 行, 移除 vendored 的 AITER `fp8_mqa_logits gfx942` 回退实现。该文件的 docstring 明确预告了删除条件 (vLLM 固定的 AITER 版本包含 ROCm/aiter#3257 修复), 本次删除说明上游条件已满足, 是临时代码到期清理的典型示例。
- `vllm/v1/attention/ops/turboquant_soa/triton_turboquant_decode_v2.py` (模块 量化算子; 类别 infra; 类型 deletion; 符号 `_tq_decode_stage1_v2`, `triton_turboquant_decode_attention_v2`) : 局部删除 572 行, 是最大的单符号删除: 移

除 `_tq_decode_stage1_v2` 整个 Triton kernel 以及配套的 import 与常量。该 kernel 包含 FP8/MSE 反量化、pair LUT、SoA 布局寻址等复杂逻辑，删除后文件保留 v1 路径，说明 v2 方案已被 unified 路径取代。

- `vllm/models/inkling/amd/ops/mm_towers.py` (模块 AMD 算子; 类别 infra; 类型 deletion; 符号 `_dmel_embed_sum_norm_kernel`, `dmel_embed_sum_norm`, `_rmsnorm_gelu_kernel`, `rmsnorm_gelu`) : 整文件删除 190 行, 移除 Inkling 视觉 / 音频 tower 的 `dmel_embed_sum_norm` 与 `rmsnorm_gelu` 两个融合算子及其 Triton kernel, 与 `lamport.py` 的删除共同构成 Inkling AMD 路径的全面收缩。
- `vllm/models/glm5next/amd/ops/kpool_compress.py` (模块 KV 压缩; 类别 infra; 类型 deletion; 符号 `compute_pooled_write_locs`, `build_pooled_page_table`) : 删除 44 行, 移除 pool id 到物理缓存 slot 映射的两个辅助函数 `compute_pooled_write_locs` 与 `build_pooled_page_table`, 说明 GLM5Next AMD 路径的 pooled KV 压缩流程已不再需要这两步预处理。
- `vllm/models/kimi_k3/nvidia/ops/cute_dsl/latent_moe_tail/primitives.py` (模块 K3 算子; 类别 infra; 类型 deletion; 符号 `block_sum_specialized`) : 删除 31 行, 移除与上游等价的 `block_sum_specialized` FP32 block 归约原语。Kimi K3 的 latent MoE tail 不再需要该归约路径, 是 K3 算子集瘦身的组成部分。
- `vllm/model_executor/layers/mamba/ops/replayssm_config.py` (模块 Mamba 配置; 类别 infra; 类型 deletion; 符号 `override_replayssm_config`) : 删除 15 行, 移除 `override_replayssm_config` `contextmanager` 及其对 `_overrides` 字典的写入入口。该工具用于临时固定 kernel launch 配置, 删除后模块仍保留查询逻辑但外部不再能注入覆盖。
- `vllm/v1/attention/ops/turboquant_soa/triton_turboquant_unified_attention.py` (模块 量化算子; 类别 infra; 类型 deletion; 符号 `_get_pit_in_query_dtype`) : 删除 15 行, 移除 `_get_pit_in_query_dtype` 及模块级 `_pit_qdtype_cache` 缓存。该缓存按 PiT 的 `data_ptr` 与目标 `dtype` 缓存类型转换结果, 删除后每次调用恢复直接转换, 消除了潜在的缓存生命周期隐患。

关键符号: `_pack_bf16_pairs`, `_unpack_bf16_pairs`, `_wait_pairs`, `_publish_input_kernel`, `_reduce_insert_kernel`, `_sconv_publish_kernel`, `_gather_norm_kernel`, `_validate_lamport_init_kernel`, `dmel_embed_sum_norm`, `_dmel_embed_sum_norm_kernel`, `rmsnorm_gelu`, `_rmsnorm_gelu_kernel`, `fp8_mqa_logits_gfx942`, `_fp8_mqa_logits_kernel`, `triton_turboquant_decode_attention_v2`, `_tq_decode_stage1_v2`, `compute_pooled_write_locs`, `build_pooled_page_table`, `block_sum_specialized`, `override_replayssm_config`, `_get_pit_in_query_dtype`

关键源码片段

`vllm/models/inkling/amd/ops/lamport.py`

整文件删除 766 行, 是本次清理中规模最大、设计最复杂的部分。该模块是基于 PDL 的死锁规避融合内核 (RS + short-conv + AG + residual + RMSNorm), 包含 8 个 Triton kernel 与完整的 liveness 论证文档, 删除它标志着 Inkling AMD 的 Lamport 路径正式退役。

以下代码为本 PR 删除内容的重组展示 (原始文件已移除), 用于理解该模块的核心设计:

被删除的 lamport.py 中两个核心位操作函数。
设计动机：payload 本身充当 Lamport 标志——用一个 32 位原子 store
同时发布两个 bf16 值，并用 0x80008000（两个负零）作为空槽哨兵。

```
@triton.jit
def _pack_bf16_pairs(values):
    """把 bf16 值打包成原子 u32 对，并保留负零作为哨兵。"""
    lo, hi = tl.split(values.reshape([values.shape[0] // 2, 2]))
    lo = lo.to(tl.uint16, bitcast=True)
    hi = hi.to(tl.uint16, bitcast=True)
    # 真实出现的负零（0x8000）在发布前改写为正零，
    # 避免与空槽哨兵冲突；消费方用 volatile 32 位加载并回写哨兵。
    lo = tl.where(lo == 0x8000, 0, lo).to(tl.uint32)
    hi = tl.where(hi == 0x8000, 0, hi).to(tl.uint32)
    return lo | (hi << 16)
```

```
@triton.jit
def _unpack_bf16_pairs(values):
    # 与 _pack_bf16_pairs 互逆：从 u32 中拆出高 / 低 16 位，
    # 按 (lo, hi) 交错还原成原始的 bf16 向量布局。
    lo = (values & 0xFFFF).to(tl.uint16).to(tl.bfloat16, bitcast=True)
    hi = (values >> 16).to(tl.uint16).to(tl.bfloat16, bitcast=True)
    return tl.interleave(lo, hi)
```

vllm/v1/attention/ops/triton_fp8_mqa_logits.py

整文件删除 262 行，移除 vendored 的 AITER fp8_mqa_logits gfx942 回退实现。该文件的 docstring 明确预告了删除条件（vLLM 固定的 AITER 版本包含 ROCm/aiter#3257 修复），本次删除说明上游条件已满足，是临时代码到期清理的典型案列。

以下为被删除文件中值得保留的硬件适配逻辑（原始文件已移除）：

```
_GFX942_CU_LDS_BYTES = 64 * 1024
# 目标两个 workgroup 同驻一个 CU，按 waves_per_eu=2 推算每个 WG 的
# LDS 预算约 28.8 KiB；0.9 安全系数给编译器的额外开销留余量。
_GFX942_PER_WG_LDS_BUDGET_BYTES = _GFX942_CU_LDS_BYTES * 9 // 20

def _gfx942_default_tile_fits_lds(num_heads: int, head_size: int) -> bool:
    """判断默认 tile (BLOCK_KV=128, num_stages=2) 是否放得进 MI300X 的 LDS。

    Triton 把 Q 保持在寄存器（循环不变量）、fp32 score 累加器放 VGPR，
    所以预期只有双缓冲的 KV tile 驻留 LDS。硬上限把 Q 和 scores 也计入
    64 KiB，防止某些 Triton 版本把它们 spill 到 LDS 后 JIT 直接报错。
    此处的保守倾向是有意的：误判只缩小 tile，漏判则是 JIT 崩溃。
    """
    BLOCK_KV = 128
    NUM_STAGES = 2
    kv_bytes = head_size * BLOCK_KV * NUM_STAGES
```

```

scores_bytes = num_heads * BLOCK_KV * 4
q_bytes = num_heads * head_size
fits_occupancy = kv_bytes < _GFX942_PER_WG_LDS_BUDGET_BYTES
fits_hardware = q_bytes + kv_bytes + scores_bytes <= _GFX942_CU_LDS_BYTES
return fits_occupancy and fits_hardware

```

vllm/v1/attention/ops/turboquant_soa/triton_turboquant_unified_attention.py

删除 15 行，移除 `_get_pit_in_query_dtype` 及模块级 `_pit_qdtype_cache` 缓存。该缓存按 PiT 的 `data_ptr` 与目标 `dtype` 缓存类型转换结果，删除后每次调用恢复直接转换，消除了潜在的缓存生命周期隐患。

以下为本 PR 删除的缓存实现（原始文件已移除），其模式本身值得记录：

```

# 被删除的 PiT 类型转换缓存。
# 设计：按 (PiT.data_ptr(), dtype) 缓存，避免每次 launcher 调用都做
# host 侧 cast；缓存键绑定 caller tensor 的 data_ptr，所有权与生命周期
# 跟随调用方，理论上同地址复用时可命中。
_pit_qdtype_cache: dict = {}

```

```

def _get_pit_in_query_dtype(PiT: torch.Tensor, qdtype: torch.dtype) -> torch.Tensor:
    key = (PiT.data_ptr(), qdtype)
    cached = _pit_qdtype_cache.get(key)
    if cached is None:
        cached = PiT.to(qdtype).contiguous()
        _pit_qdtype_cache[key] = cached
    return cached

```

评论区精华

本 PR 的 review 过程极为精简，没有实质性的技术讨论：

- claude[bot] 发表了一条自动评论，提示仓库配置的是人工 review 模式，可订阅 @claude review。
- sfeng33 直接给出 APPROVED，无评论文本。
- 4 条 Issue 评论全部是 `/ci run` 命令与 Buildkite 触发确认，不涉及代码讨论。

因此本 PR 的关键决策（哪些代码算 dead、删除边界如何划定、AITER 上游修复是否已落地）完全由作者与审批人依据代码事实判断，没有留下可追溯的讨论记录。对于维护者而言，这类大规模删除缺少 review 讨论是一个值得注意的信号，后续如出现回归，排查将缺少上下文。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险集中在死代码判断的准确性与 GPU 专有路径的验证覆盖上：
 1. 隐性引用风险：source_summary 显示没有明显的源码 / 测试联动信号，说明引用检查大概率基于静态搜索。若存在动态导入、字符串拼接模块名或未纳入搜索范围的间接调用（如

importlib.import_module 或模型注册表内的延迟加载），整文件删除（尤其是 lamport.py 这类 docstring 中明确写了 "The public integration surface isLamportRSConv.rs_sconv_ag_add_norm" 的模块）会直接导致运行时 ImportError。

2. GPU 专有路径缺少专项测试：被删代码集中在 AMD gfx942/MI300X 与 Blackwell 量化路径，这些路径在通用 CI 中往往只有编译级覆盖。TurboQuant decode v2 删除 572 行后，若 v1 路径与 unified 路径存在共享状态或残留引用，在特定硬件配置下才会暴露。
3. 缓存与配置清理的连带影响：replayssm_config.py 删除 override_replayssm_config 后 _overrides 字典仍在但修改入口消失，若下游测试仍以某种方式调用该 contextmanager，会得到 NameError；_pit_qdtype_cache 的删除则消除了潜在的内存驻留问题，属于正向风险消除。
4. 回归成本：1895 行删除后，若未来某个被删 kernel 需要恢复（例如 Inkling AMD 路径重新启用 Lamport 融合内核），需要从 git 历史中找回，这类恢复成本是纯删除类 PR 的固有代价。
 - 影响：影响范围集中在算子层与模型支持层：
 - 对用户：无可见行为变化，vLLM 对外 API 与推理结果均不受影响。删除的死代码不影响任何受支持的模型路径。
 - 对系统：代码库净减少 1895 行，主要收益是降低 Inkling、GLM5Next、Kimi K3、TurboQuant 与 Mamba 相关目录的检索与维护成本；triton_fp8_mqa_logits.py 的删除同时消除了一个含 TODO 的临时 vendored 模块，避免后续维护者误以为仍需保留。
 - 对团队：对 AMD/ROCm 内核维护者是一个信号——AITER 上游已补齐 gfx942 修复、旧的 fallback 不再需要；对模型团队来说，Inkling AMD 的 Lamport 与 mm_towers 算子被整体移除，说明对应的融合路径已经不在当前支持的模型执行流中。无部署与配置配套改动。
 - 风险标记：大规模删除需回归验证，GPU 专有路径缺少专项测试，依赖隐性引用不存在的假设，无 review 讨论记录

关联脉络

- PR #56254 [DSA] Wire DeepGEMM sparse MQA logits into the DeepSeek V4.1 indexer: 被删的 triton_fp8_mqa_logits.py 正是 DSv4.1 sparse indexer 在 gfx942 上的临时回退（docstring 提到 NUM_HEADS=64/HEAD_SIZE=128 的 DSv4 sparse indexer 场景），该 PR 引入的 DeepGEMM sparse MQA logits 路径成熟后，vendored fallback 便失去存在价值。
- PR #56743 [ROCm][Perf] Optimize DSv4.1 K=512 decode top-k on gfx950: 同为 ROCm 路径的 kernel 优化工作，说明 AMD 算子路径仍在活跃演进；本 PR 删除的多个 gfx942 相关 fallback 与优化路径的交替更新，共同构成 AMD kernel 的持续瘦身脉络。
- PR #56346 [Perf][Kernel] Add sampled filtering for persistent top-k: 同属 kernel 层清理与优化的演进而非；top-k 采样过滤与本次删除的 fp8_mqa_logits 都服务于稀疏注意力/MQA 打分链路，反映该链路的实现正在被更新方案逐步替换。