PR #43135 完整报告

vllm-project/vllm

[Perf][gpt-oss] Downgrade triton_kernels to v3.5.1

合并时间：2026-05-21 05:13

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43135

执行摘要

- 一句话：降级 triton_kernels 到 v3.5.1 恢复 GPT-OSS 解码性能
- 推荐动作：此 PR 值得精读，尤其是 triton_kernel_moe_forward 中的路由分支选择（if use_legacy_triton_kernels and expert_map is None）展示了如何在不破坏新 API 的前提下回滚关键优化。设计上通过 import 时判断 API 版本来选择路径，是一种干净的兼容性模式。未来需要关注上游修复后如何平滑过渡回 v3.6.0+。

功能与动机

PR 正文指出，triton_kernels v3.5.1 → v3.6.0 的升级（#30525）以及路由重写（#38504）导致 GPT-OSS 模型解码性能从 307.4 tok/s 降至 230.1 tok/s。本 PR 旨在恢复该性能，同时保持对 v3.6.0+ 的兼容。上游性能回归已追踪至 triton-lang/triton#9969。

实现拆解

1. 降级 triton_kernels 版本：修改 cmake/external_projects/triton_kernels.cmake，将默认标签从 v3.6.0 改为 v3.5.1，确保构建时拉取旧版本。
2. 放宽 legacy 路径触发条件：在 gpt_oss_triton_kernels_moe.py 的启动块中，原先仅在 ROCm 平台当 SparseMatrix 导入失败时设置 use_legacy_triton_kernels = True，现在改为在所有平台上当导入失败时均激活 legacy 路径，并添加注释说明这是临时候选，等待上游修复（triton-lang/triton#9969）。
3. 优化无 EP 时的路由：在 triton_kernel_moe_forward 函数中，当 use_legacy_triton_kernels 为 True 且 expert_map 为 None 时，直接调用 triton_kernels.routing.routing() 融合内核（一次 launch）完成 softmax、topk、bitmatrix 打包和路由元数据构建；否则回到通用路径（topk + pack_bitmatrix + routing_from_bitmatrix 三次 launch），通过 make_routing_data 组织数据。
4. 兼容垫片：保留 v3.6.0+ 路径（通过 _patch_make_bitmatrix_metadata 修补 SparseMatrix 的元数据函数），确保使用系统级安装的 triton_kernels v3.6.0+ 时仍能正常工作。
5. 测试调整：在 tests/kernels/quantization/test_mxfp4_triton_ep.py 中移除 pytest 参数化，仅保留 test_expert_map_remap 用例，聚焦 EP 路径的正确性验证（因为非 EP 路径改为使用融合路由，mock 不再适用）。

关键文件：

- vllm/model_executor/layers/fused_moe/experts/gpt_oss_triton_kernels_moe.py（模块 MoE 路由；类别 source；类型 core-logic；符号 use_legacy_triton_kernels, triton_kernel_moe_forward）：核心逻辑变更：实现 triton_kernels 版本选择、legacy 路径激活以及融合路由优化。
- tests/kernels/quantization/test_mxfp4_triton_ep.py（模块 路由测试；类别 test；类型 test-coverage；符号 test_routing_path_selection, test_expert_map_remap）：测试适配：移除非 EP 路径的 mock 测试，专注验证 EP 下的 expert_map 重映射逻辑。
- cmake/external_projects/triton_kernels.cmake（模块 构建配置；类别 infra；类型 configuration）：构建依赖版本控制：将默认 triton_kernels 标签从 v3.6.0 改为 v3.5.1，是降级的直接执行点。

关键符号：triton_kernel_moe_forward

关键源码片段

vllm/model_executor/layers/fused_moe/experts/gpt_oss_triton_kernels_moe.py

核心逻辑变更：实现 triton_kernels 版本选择、legacy 路径激活以及融合路由优化。

```
# Two API generations of triton_kernels are supported:
# - v3.5.1 (the version bundled with vLLM): exposes `routing()` and
# `routing_from_bitmatrix()` in triton_kernels.routing; the `Bitmatrix`
# constructor takes a `scratchpad` argument.
# - v3.6.0+: removes the `routing` module in favor of a `SparseMatrix`
# based path, and adds a `dtype=BIT` kwarg to `Bitmatrix`. Used only
# when the user has triton_kernels installed system-wide at v3.6.0+.
#
# `use_legacy_triton_kernels` selects between them at import time based on
# whether `SparseMatrix` is importable.
use_legacy_triton_kernels = False

if has_triton_kernels():
    try:
        import triton_kernels.swiglu
        from triton_kernels.matmul_ogs import (
            FnSpecs,
            FusedActivation,
            GatherIndx,
            RoutingData,
            ScatterIndx,
            matmul_ogs,
        )
        from triton_kernels.tensor import (
            BIT,
            Bitmatrix,
        )

        try:
```

```
            from triton_kernels.tensor import (
                SparseMatrix,
                make_ragged_tensor_metadata,
            )
        except ImportError:
            # TODO(mgoin): drop the v3.5.1 pin and remove this fallback once
            # the gpt-oss perf regression in v3.6.0+ is resolved upstream.
            # Tracking: https://github.com/triton-lang/triton/issues/9969
            use_legacy_triton_kernels = True
        if not use_legacy_triton_kernels:
            _patch_make_bitmatrix_metadata()
    except (AttributeError, ImportError) as e:
        logger.error(
            "Failed to import Triton kernels. Please make sure your triton "
            "version is compatible. Error: %s",
            e,
        )
```

评论区精华

- 融合路由路径作用域限制：gemini-code-assist[bot] 指出，新优化的 routing() 融合路径仅在 expert_map 为 None 时生效，即仅适用于非专家并行（EP）配置。EP 场景仍采用三次内核启动，性能缺口未解决。PR 最终被批准，表明维护者接受此范围限制。
- 代码重复问题：同一 reviewer 注意到 else 块中的通用路径代码与 make_routing_data 函数高度重复，建议重构。PR 未进行重构即被合并，团队可能倾向于在后续独立清理。
  - 融合路由路径仅在无 EP 时激活 (performance): PR 被批准，团队未对此进行额外修改，表明当前范围可接受。
  - else 块代码重复 (design): PR 被批准，未重构，团队接受当前重复。

风险与影响

- 风险：
  1. 性能缺口（EP 场景）：只有非 EP 配置能享受融合路由加速，EP 配置仍然使用旧的三次内核启动路径，性能可能仍低于 v3.5.1 基线。
  2. 代码重复与维护风险：triton_kernel_moe_forward 中的通用路径与 make_routing_data 内部逻辑重复，后续如果修改一方可能不同步，导致行为不一致。
  3. 上游依赖风险：降级锁定 triton_kernels 为 v3.5.1，可能错过 v3.6.0 及更高版本的 bug 修复和新特性。当上游修复性能回归后，需要主动升级并移除兼容垫片。
  4. 测试覆盖不足：现有测试仅通过 mock 验证逻辑路径，缺少端到端性能基准测试和一致性测试，可能遗漏运行时数值错误。 - 影响：用户影响：使用 GPT-OSS 模型的用户将在解码阶段获得明确性能恢复（输出吞吐从 230.1 tok/s 回升至 307.4 tok/s）。非 GPT-OSS 模型或无 EP 的配置也可能受益于融合路由优化。对于使用 EP 的 GPT-OSS 用户，性能可能仍不如严格绑定的 v3.5.1，但 PR 保留了 v3.6.0+ 兼容性。系统影响：构建系统默认拉取 triton_kernels v3.5.1，但若用户已系统级安装 v3.6.0+，仍可通过 export use_legacy_triton_kernels 等机制运行（具体取决于环境）。团队影响：维护者

需跟踪上游 issue #9969，待修复后撤销降级并清理兼容代码。

- 风险标记：性能缺口（EP 场景），代码重复技术债务，降级可能遗漏上游修复，测试以 mock 为主缺乏端到端覆盖

关联脉络

- PR #30525 Bump triton_kernels to v3.6.0: 该 PR 升级了 triton_kernels 版本，是引入性能回归的根源之一。
- PR #38504 Rewrite gpt-oss routing path: 该 PR 重写了路由逻辑，丢弃了融合路由内核，是引入性能回归的另一原因。
- PR #39236 Alternative downgrade PR: 该 PR 是备选方案，采取了更小 diff 的方式恢复性能，但本 PR 选择了 deprecate v3.6.0 路径并保留兼容垫片。