

PR #54040 完整报告

vllm-project/vllm

[Kernel] Retire the DSv3 router GEMM CUDA kernel

合并时间: 2026-08-31 15:15

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/54040>

执行摘要

- 一句话: 退役 DSv3 专用 router GEMM 内核, 删除千余行 CUDA 代码
- 推荐动作: 值得精读。该 PR 展示了「专用内核被通用内核替代后退役」的典型清理范式: 先由 `ll_bf16_gemm` 覆盖性能场景, 再删除冗余 fallback 与配套调优矩阵。阅读时建议重点关注两处设计决策: 一是 GateLinear 分层分发的可维护性设计 (每层独立 eligibility 标志, 删除成本低); 二是 `fp32_router_gemm` 形状矩阵的连带收窄是否会对 GLM-5.2 的实际部署性能产生可感知影响, 建议在合入后观察 GLM-5.2 的 router 延迟监控。

功能与动机

PR body 未填写实质内容, 动机需从代码脉络推断: `gate_linear.py` 注释显示 DSv3 专用内核是作为 cuteDSL `ll_bf16_gemm` 的 fallback 引入的 (注释引用 vllm-project/vllm PR#44217), 而 cuteDSL 路径已成熟稳定。专用内核长期处于兜底位置却承担着约 850 行模板实例化与编译负担, 退役它可以统一路由路径、降低维护成本; 同时 `fp32_router_gemm.cu` 中 (6144, 256) 的 tuned 几何与 DSv3 内核共享调优矩阵, 一并清理。

实现拆解

按以下 5 步拆解实现:

1. 删除 CUDA 内核层: 移除 `csrc/libtorch_stable/moe/dsv3_router_gemm_entry.cu` (210 行入口, 含 `LoopUnroller` 模板递归分派与 `dsv3_router_gemm` 实现)、`dsv3_router_gemm_bf16_out.cu` 与 `dsv3_router_gemm_float_out.cu` (各 337 行的 `router_gemm_kernel_bf16_output` / `router_gemm_kernel_float_output` 内核及 1-16 token 的全量模板实例化)。同时在 `moe_ops.h` 删除声明、`csrc/libtorch_stable/moe/torch_bindings.cpp` 删除 `m.def("dsv3_router_gemm...")` 绑定, 并从 `CMakeLists.txt` 移除对应编译项。
2. 删除 Python 算子接口: `vllm/_custom_ops.py` 中直接移除 `dsv3_router_gemm()` 包装函数 (约 15 行), 保留 `fp32_router_gemm` 及其 `register_fake` 注册不动。
3. 收窄 GateLinear 分发契约: `vllm/model_executor/layers/fused_moe/router/gate_linear.py` 删除 `DSV3_SUPPORTED_NUM_EXPERTS`、`DSV3_SUPPORTED_HIDDEN_SIZES`、`DSV3_UNSUPPORTED_SHAPES`、`allow_dsv3_router_gemm`、`_dsv3_max_batch` 等属性与 `forward` 中的 Tier 2 分支; `FP32_SUPPORTED_SHAPES` 从 `{(3072,256), (6144,128), (6144,256)}` 收窄为 `{(3072,256), (6144,128)}`, `forward` 分发链相应重编号

为 5 层。

4. 连带清理 `fp32_router_gemm` 形状: `csrc/libtorch_stable/fp32_router_gemm.cu` 删除 GLM-5.2 (E=256, H=6144) 的 B300 wide-block/SM90 分支及 `INstantiateAll` 实例化, `fp32_router_gemm_entry.cu` 同步移除该形状的入口登记。
5. 配套基准与测试: `benchmarks/kernels/benchmark_router_gemm.py` 移除 `DSV3_SUPPORTED_*` 常量与 `vllm provider` 中的 `ops.dsv3_router_gemm` 分支; `tests/kernels/test_fp32_router_gemm.py` 将 `SHAPES` 收窄为 `[(3072, 256), (6144, 128)]`, 不再覆盖 `(6144, 256)`。

关键文件:

- `vllm/model_executor/layers/fused_moe/router/gate_linear.py` (模块 MoE 路由; 类别 source; 类型 data-contract; 符号 `GateLinear`, `GateLinear.forward`, `GateLinear.init`, `FP32_SUPPORTED_SHAPES`): MoE 路由核心分发入口, 删除 DSV3 专用内核的 `eligibility` 判定与 `forward Tier 2` 分支, 并将 `FP32_SUPPORTED_SHAPES` 收窄为 2 个形状, 是本次行为契约变化的核心。
- `vllm/_custom_ops.py` (模块 算子注册; 类别 source; 类型 core-logic; 符号 `dsv3_router_gemm`, `fp32_router_gemm`): 删除 `dsv3_router_gemm` Python 包装函数, 彻底切断 Python 侧对已退役内核的调用入口; `fp32_router_gemm` 及其 `fake` 注册保留。
- `csrc/libtorch_stable/moe/dsv3_router_gemm_entry.cu` (模块 CUDA 内核; 类别 other; 类型 deletion; 符号 `dsv3_router_gemm`, `LoopUnroller`): DSV3 内核的 C++ 入口实现 (210 行), 含 `LoopUnroller` 模板递归分派、形状校验与 SM90 检查, 是本次退役的核心删除对象。
- `csrc/libtorch_stable/moe/dsv3_router_gemm_bf16_out.cu` (模块 CUDA 内核; 类别 other; 类型 deletion; 符号 `router_gemm_kernel_bf16_output`, `invokeRouterGemmBf16Output`): DSV3 内核 bf16 输出变体 (337 行), 含 `router_gemm_kernel_bf16_output` 内核与 1-16 token 全量模板实例化, 随入口一并删除。
- `csrc/libtorch_stable/moe/dsv3_router_gemm_float_out.cu` (模块 CUDA 内核; 类别 other; 类型 deletion; 符号 `router_gemm_kernel_float_output`, `invokeRouterGemmFloatOutput`): DSV3 内核 fp32 输出变体 (337 行), 与 bf16 变体对称, 删除后不再保留任何 DSV3 专用 CUDA 实现。
- `csrc/libtorch_stable/moe/moe_ops.h` (模块 算子接口; 类别 source; 类型 core-logic): 移除 `dsv3_router_gemm` 函数声明, 保持头文件与实现同步清理。
- `csrc/libtorch_stable/moe/torch_bindings.cpp` (模块 算子接口; 类别 source; 类型 core-logic): 删除 `m.def("dsv3_router_gemm...")` 的 `TORCH_LIBRARY` 注册, 避免算子表残留无效条目。
- `csrc/libtorch_stable/fp32_router_gemm.cu` (模块 CUDA 内核; 类别 other; 类型 entrypoint; 符号 `invokeFp32RouterGemm`): 连带移除 GLM-5.2 (E=256, H=6144) 的 B300 tuned wide-block 几何与模板实例化, 缩小 fp32 内核维护面, 但也构成 GLM-5.2 的行为变化点。
- `benchmarks/kernels/benchmark_router_gemm.py` (模块 基准脚本; 类别 source; 类型 entrypoint): 基准脚本同步移除 DSV3 形状常量与 `ops.dsv3_router_gemm` 分支, 避免基准路径引用已删除接口。

- tests/kernels/test_fp32_router_gemm.py (模块 内核测试; 类别 test; 类型 test-coverage) : 测试覆盖同步收窄: SHAPES 移除 (6144, 256), 与 fp32 内核的实际实例化保持一致。
- CMakeLists.txt (模块 构建配置; 类别 docs; 类型 documentation) : 从构建清单移除已删除的 3 个 CUDA 源文件, 保证编译系统与源码一致。

关键符号: GateLinear.forward, GateLinear.init, fp32_router_gemm, invokeFp32RouterGemm, dsv3_router_gemm

关键源码片段

vllm/model_executor/layers/fused_moe/router/gate_linear.py

MoE 路由核心分发入口, 删除 DSV3 专用内核的 eligibility 判定与 forward Tier 2 分支, 并将 FP32_SUPPORTED_SHAPES 收窄为 2 个形状, 是本次行为契约变化的核心。

```
# SPDX-License-Identifier: Apache-2.0
@PluggableLayer.register("gate_linear")
class GateLinear(ReplicatedLinear):
    """MoE gate linear layer with multi-tier GEMM dispatch:

    1. cuteDSL ll_bf16_gemm (SM90+, M<=16, bf16 in, fp32 out, K divisible by 8)
    2. fp32 specialized kernel (SM90+, bf16/fp32 in, fp32 out, M<=32,
       (H, E) in {(3072, 256), (6144, 128)})
    3. experimental bf16x3 CuteDSL kernel (opt-in, SM100, bf16 in, fp32 weight)
    4. cuBLAS bf16x3bf16→fp32 (SM90+ + bf16 weight + fp32 out_dtype)
    5. F.linear via ReplicatedLinear (ultimate fallback)
    """

    # 仍保留实例化的 fp32 专用内核形状:
    # (3072, 256) -> MiniMax-M2/M2.5, (6144, 128) -> MiniMax-M3
    # GLM-5.2 的 (6144, 256) 已随 DSV3 内核一并退役。
    FP32_SUPPORTED_SHAPES = {(3072, 256), (6144, 128)}
    FP32_MAX_TOKENS = 32

    def forward(
        self, x: torch.Tensor
    ) -> torch.Tensor | tuple[torch.Tensor, Parameter | None]:
        # Tier 1: cuteDSL ll_bf16_gemm (SM90+, M<=16, bf16 输入)
        if self.allow_ll_bf16_gemm and x.shape[0] <= 16 and x.dtype == torch.bfloat16:
            from vllm.model_executor.kernels.linear.cute_dsl.ll_bf16 import (
                ll_bf16_gemm,
            )

            output = ll_bf16_gemm(x, self.weight)
            return output, None

        # Tier 2: fp32 专用内核 (H=3072/E=256、H=6144/E=128, M<=32) 。
        # 分发被包进自定义 op, 让 torch.compile/CUDA-graph 捕获时
```

```

# 不会冻结运行时的 num_tokens 分支。
if self.allow_fp32_router_gemm and x.dtype in (
    torch.float32,
    torch.bfloat16,
):
    output = torch.ops.vllm.fp32_router_gemm_dispatch(
        x, self.weight, self.allow_bf16x3_router_gemm
    )
    return output, None

# Tier 3: 实验性 bf16x3 CuteDSL 内核, 用于 fp32 router 权重 (opt-in)
if self.allow_bf16x3_router_gemm and x.dtype == torch.bfloat16:
    from vllm.model_executor.layers.fused_moe.router.bf16x3_router_gemm_cutedsl import
    ( # noqa: E501
        bf16x3_router_gemm,
    )

    output = bf16x3_router_gemm(x, self.weight)
    return output, None

# Tier 4: cuBLAS bf16→fp32。DSV3 专用内核退役后,
# DeepSeek-V3/Kimi-K2 在 cuteDSL 不可用时会直接落到这里,
# 由 torch.mm 的 out_dtype epilogue 折叠 fp32 转换。
if self.allow_cublas_router_gemm and x.dtype == torch.bfloat16:
    output = torch.mm(x, self.weight.T, out_dtype=torch.float32)
    return output, None

# Tier 5: F.linear (ReplicatedLinear) 最终兜底
if self.out_dtype is not None and x.dtype != self.weight.dtype:
    x = x.to(self.weight.dtype)
output, output_bias = super().forward(x)
if self.out_dtype is not None and output.dtype != self.out_dtype:
    output = output.to(self.out_dtype)
return output, output_bias

```

vllm/_custom_ops.py

删除 `dsv3_router_gemm` Python 包装函数, 彻底切断 Python 侧对已退役内核的调用入口;
`fp32_router_gemm` 及其 fake 注册保留。

```

# 删除 dsv3_router_gemm 后的 Python 算子入口:
# fp32_router_gemm 直接走 torch.ops._C 的旧式绑定, 并保留 fake 注册,
# 供 torch.compile 在符号执行时使用。
def fp32_router_gemm(
    hidden_states: torch.Tensor,
    router_weight: torch.Tensor,
) -> torch.Tensor:
    output = torch.empty(
        hidden_states.shape[0],
        router_weight.shape[0],

```

```

        device=hidden_states.device,
        dtype=torch.float32,
    )
    torch.ops._C.fp32_router_gemm(output, hidden_states, router_weight)
    return output

```

```

if hasattr(torch.ops, "_C") and hasattr(torch.ops._C, "fp32_router_gemm"):

```

```

@register_fake("_C::fp32_router_gemm")
def fp32_router_gemm_fake(
    output: torch.Tensor,
    mat_a: torch.Tensor,
    mat_b: torch.Tensor,
) -> None:
    return

```

benchmarks/kernels/benchmark_router_gemm.py

基准脚本同步移除 DSV3 形状常量与 `ops.dsv3_router_gemm` 分支，避免基准路径引用已删除接口。

```

# 基准脚本 head 版本：DSV3 分支移除后，vllm provider 只对比
# fp32 专用内核、gpt-oss 内核与 F.linear 兜底。
if provider == "vllm":

```

```

def runner():
    if allow_fp32_router_gemm:
        ops.fp32_router_gemm(mat_a, mat_b)
    elif allow_gpt_oss_router_gemm:
        ops.gpt_oss_router_gemm(mat_a, mat_b, bias)
    elif is_fp32_router_model:
        # batch_size > FP32_MAX_TOKENS: fall back to F.linear
        F.linear(mat_a.float(), mat_b)
    else:
        F.linear(mat_a, mat_b)

```

评论区精华

本 PR 没有实质性的 review 讨论：[zyongye](#) 与 [Harry-Chen](#) 均直接 APPROVED，[review_comments_count](#) 为 0。评论区内只有 5 轮 `/ci run` 触发记录（Buildkite CI #85938、#85970、#86119、#86176、#86265、#86281），覆盖了 6 个 commit 的持续集成验证，且所有 CI 均通过。从代码注释可以还原的设计共识是：DSV3 专用内核的角色只是 cuteDSL 的 fallback，且此前已有 PR#44217 确立其适用范围与 `_dsv3_max_batch`（Hopper 16 / Blackwell 8）的限制，本次退役是在该前提下的自然收敛。

- Review 与 CI 流程 (other): 无未解决问题，评审结论为通过。

风险与影响

- 风险:

1. 核心推理路径变更: `gate_linear.py` 删除 `allow_dsv3_router_gemm` 分支后, DeepSeek-V3/Kimi-K2 (H=7168, E=256/384) 的路由 GEMM 在 `ll_bf16_gemm` 不可用 (如 `is_available()` 为 `False`、K 不满足整除条件) 时直接落到 Tier 4 cuBLAS `torch.mm(..., out_dtype=torch.float32)`, 正确性不变但延迟可能略增。
2. GLM-5.2 形状连带移除: `fp32_router_gemm` 删除 (6144, 256) 形状后, GLM-5.2 的 `fp32 router` 权重不再有专用内核, 将回退到 `F.linear` 或 `bf16x3` (仅 SM100 opt-in)。这是 PR 中容易被忽略的行为变化, PR 描述未单独说明。
3. 接口残留风险: 若仓库内仍有未发现的 `ops.dsv3_router_gemm` 或 `torch.ops._moe_C.dsv3_router_gemm` 引用, 运行时会直接 `AttributeError`; `benchmark` 与测试已同步清理, 但 `e2e` 之外的第三方扩展不在覆盖范围内。
4. 测试覆盖缺口: `tests/kernels/test_fp32_router_gemm.py` 只删除了 (6144, 256) 用例, 未新增针对 DeepSeek-V3 路由路径等价性的专门回归, DSV3 形状的正确性依赖已有 `e2e` 与 `ll_bf16_gemm` 测试间接保障。
 - 影响: 对用户与部署的影响集中在 DeepSeek-V3、Kimi-K2、GLM-5 系列模型: 正常场景下 DeepSeek-V3/Kimi-K2 的路由计算由 `cuteDSL` 承接, 性能基本持平; GLM-5.2 的 `fp32 router` 场景则可能因失去专用内核而出现性能回退。对工程与团队的影响是正向的: CUDA 编译面减少约 1000 行模板代码, 二进制体积与编译时间下降, 后续 `router GEMM` 调优只需聚焦 `cuteDSL` 与 `fp32` 两条主线, 降低了内核维护负担。影响范围限定在 NVIDIA CUDA 后端与 MoE 路由层, 不涉及 ROCm/XPU/CPU 路径。
 - 风险标记: 核心路由路径变更, GLM-5.2 形状连带移除, 依赖 `cuteDSL` 可用性, DSV3 形状回归依赖 `e2e`

关联脉络

- PR #44217 原始引入 DSV3 router GEMM 的 PR (标题未在上下文中提供):
`gate_linear.py` 的注释明确引用该 PR 作为 DSV3 内核适用范围与 `_dsv3_max_batch` 限制的依据; 本 PR 即是对该内核的退役收尾。