PR #41255 完整报告

vllm-project/vllm

[Perf] Intergrate Tile Kernels `head_compute_mix_kernel` for Deepseek-V4

合并时间：2026-05-02 03:01

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41255

执行摘要

- 一句话：集成 TileLang 融合核优化 DeepSeek-V4 的 hc_head 计算，减少内存开销，提升吞吐。
- 推荐动作：值得精读，尤其适合希望了解 TileLang 内核集成流程和写融合核替换 PyTorch 多步操作的工程师。该 PR 展示了设计权衡（参数调优、与现有代码对齐）和性能验证方法。

功能与动机

参考 Issue #40902 DeepSeek V4 Roadmap 中 Kernel Integration 部分，目标是集成 TileKernels 中的 head_compute_mix_kernel 以降低 hc_head 操作的计算和内存开销。原实现通过多次 Torch 操作产生大量中间张量，新核将其融合为一个自定义操作。

实现拆解

实现步骤

1. 定义 TileLang 融合核：在 vllm/model_executor/layers/mhc.py 中新增 hc_head_fuse_tilelang，这是一个使用 @tilelang.jit 装饰的两遍计算核。第一遍跨残差通道累积每个 token 的平方和（用于 RMS 归一化）以及 hc_mult 个空间点积（与 fn 矩阵行的投影）；第二遍使用 sigmoid 门控加权融合各通道到输出。全部在片上和共享内存完成，不写出中间变量。
2. 注册自定义操作：在 mhc.py 中实现包装函数 _hc_head_fused_kernel，该函数调用 TileLang 核，然后通过 torch.library.custom_op 将其注册为 torch.ops.vllm.hc_head_fused_kernel，使 PyTorch 编译器可以识别和调度。
3. 修改模型调用点：在 vllm/model_executor/models/deepseek_v4.py 中重写 hc_head 函数，从原来的多步 torch 操作（x.square().mean(), F.linear, sigmoid, sum）替换为：将输入 reshape 为 [num_tokens, hc_mult, hidden_size]，直接调用自定义操作，最后 reshape 回原始形状。同时移除了不再需要的 import torch.nn.functional as F。
4. 性能验证：PR body 提供 SPEED-Bench 对比数据，显示显著性能提升。审核者 zyongye 也在 B300 GPU 上本地验证了准确度（GPQA 90 分）。

关键文件：

- vllm/model_executor/layers/mhc.py（模块 MHC 层；类别 source；类型 core-logic；符号 hc_head_fuse_tilelang, _hc_head_fused_kernel）：核心变更文件，新增 TileLang 融合核定义和自定义操作注册，是性能优化的主体。

- vllm/model_executor/models/deepseek_v4.py（模块 前向入口；类别 source；类型 data-contract）：模型调用点改动，将 hc_head 函数替换为自定义操作调用，直接影响前向性能。

关键符号：hc_head_fuse_tilelang, _hc_head_fused_kernel, hc_head

关键源码片段

vllm/model_executor/models/deepseek_v4.py

模型调用点改动，将 hc_head 函数替换为自定义操作调用，直接影响前向性能。

```
@torch.compile(backend=current_platform.simple_compile_backend)
def hc_head(
    hidden_states: torch.Tensor,
    hc_fn: torch.Tensor,
    hc_scale: torch.Tensor,
    hc_base: torch.Tensor,
    rms_norm_eps: float,
    hc_eps: float,
) -> torch.Tensor:
    # 原实现：x = x.flatten(1).float(); rsqrt=...; mixes=F.linear(x,hc_fn)*rsqrt; ...
    # 新实现：直接调用融合核，避免中间张量
    hc_mult, hidden_size = hidden_states.shape[-2:]
    outer_shape = hidden_states.shape[:-2]
    hs_flat = hidden_states.view(-1, hc_mult, hidden_size)
    num_tokens = hs_flat.shape[0]
    out = torch.empty(
        num_tokens, hidden_size, dtype=torch.bfloat16, device=hidden_states.device
    )
    torch.ops.vllm.hc_head_fused_kernel(
        hs_flat, hc_fn, hc_scale, hc_base, out,
        hidden_size, rms_norm_eps, hc_eps, hc_mult
    )
    return out.view(*outer_shape, hidden_size)
```

评论区精华

- 魔法数字 h_block 的讨论：gemini-code-assist[bot] 建议将 math.gcd(512, hidden_size) 中的 512 提取为命名常量以提升可维护性。zyongye 追问 512 的来源。Isotr0py 回应：该值与现有 mhc_post_tilelang 核中 h_blk 一致，且实验比较 1024/128 和 512/64 组合性能差异不大，因此为保持对齐而沿用。
- 准确度验证：zyongye 批准 PR 并说明本地测试 GPQA 得分 90，验证了准确度保持。
  - h_block 魔法数讨论 (design): 保持与现有代码一致，使用 1024 作为默认 h_blk，最终通过 gcd 得到 512。

风险与影响

- 风险：

1. 编译依赖：新核依赖 TileLang 编译器和运行库，若环境未正确安装 TileLang 或 GPU 不支持（如 AMD ROCm 或旧架构），会导致模型加载失败。
2. 硬编码参数：核中 n_thr=128, h_blk=1024（最终取 gcd 后实际为 512）是针对 4xGB200 调优的，其他 GPU 架构可能需要重新调优以获得最佳性能。
3. 模型前向路径变更：hc_head 是 DeepSeek-V4 模型前向的关键部分，任何数值错误都会影响生成质量。虽然审核者验证了准确度，但缺少单元测试（无测试文件改动）。
4. 回归风险：移除了 torch.nn.functional 导入，但原代码可能在其他地方间接依赖。 - 影响：用户：受益于 ~9% 请求吞吐提升和 ~11% TPOT 降低，对高并发推理场景有显著改善。系统：融合核减少了全局内存访问和 kernel launch 次数，降低了 GPU 内存压力和调度开销。团队：在 DeepSeek V4 Performance 路线图中迈出重要一步，为后续类似 TileKernels 集成提供了可复用的模式。

- 风险标记：引入 TileLang 编译依赖，硬编码核参数需调优，缺少自动化测试，改变模型前向关键路径

关联脉络

- PR #40902 [Roadmap] DeepSeek V4: 该 PR 是路线图上 Kernel Integration 环节的一部分，直接关联 Issue。
- PR #41443 [DSV4] Add knob to enable pre-attn gemm: 同期对 DeepSeek V4 性能优化，涉及多流 GEMM。
- PR #41441 AR+mhc_post fusion (未在历史列表中但 Issue 提及): 直接相关，针对 mhc_post 的融合，与 hc_head 同属 MHC 模块。