PR #40687 完整报告

vllm-project/vllm

[ROCm][Perf] Support N=5 in wvSplitK skinny GEMM kernels for speculative decoding

合并时间：2026-05-29 00:28

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40687

执行摘要

- 一句话：ROCm 瘦 GEMM 内核支持 N=5，加速推测解码验证
- 推荐动作：值得合并的针对性性能优化。建议未来考虑自动化特化更多 N 值的方法，以减少手动添加 *case* 的工作量和编译时间。同时可关注 custom op 的优化机会。

功能与动机

在 AMD ROCm 平台上使用推测解码（如 Eagle3， num_speculative_tokens=4）时，目标模型验证阶段的 batch size 为 5（1 个原始 token + 4 个推测 token）。wvSplitK 瘦 GEMM 内核原本仅支持 N<=4，导致验证步骤 fallback 到 torch.nn.functional.linear（hipBLAS），性能不佳。PR 旨在通过扩展内核支持，使验证步骤也能使用高性能 HIP 内核，从而提升推测解码的整体性能。

实现拆解

1. 修改 dispatch 阈值：在 vllm/model_executor/layers/utils.py 的 rocm_unquantized_gemm_impl 函数中，将 wvSplitK 内核的启用条件从 0 < n <= 4 改为 0 < n <= 5，使 batch size 为 5 的验证操作也能使用该内核。
2. 添加内核 tile 配置：在 csrc/rocm/skinny_gemms.cu 的 wvSplitK 函数 switch 语句中添加 *case* 5: 分支，根据是否使用 wave32 模式设置相应的 tile 配置（32x16 或 64x16），确保内核能正确处理 N=5 的矩阵乘法。

关键文件：

- vllm/model_executor/layers/utils.py（模块 GEMM 调度器；类别 source；类型 data-contract；符号 rocm_unquantized_gemm_impl）：修改了 wvSplitK 内核的 dispatch 条件，将 N 上限从 4 提升到 5，是 PR 性能提升的关键决策点。
- csrc/rocm/skinny_gemms.cu（模块 HIP 内核；类别 other；类型 core-logic；符号 wvSplitK）：添加了 N=5 的 tile 配置，使内核能处理新的 batch size。

关键符号：rocm_unquantized_gemm_impl, wvSplitK

关键源码片段

vllm/model_executor/layers/utils.py

修改了 wvSplitK 内核的 dispatch 条件，将 N 上限从 4 提升到 5，是 PR 性能提升的关键决策点。

```
# vllm/model_executor/layers/utils.py ( 修改 dispatch 条件 )
def rocm_unquantized_gemm_impl(...):
    # ... 前面的条件判断 ...

    use_skinny = (
        envs.VLLM_ROCM_USE_SKINNY_GEMM
        and (on_gfx9() or on_gfx1x())
        and x.dtype in [torch.float16, torch.bfloat16]
        and k % 8 == 0
    )

    if use_skinny:
        x_view = x.reshape(-1, x.size(-1))
        # 原条件 0 < n <= 4，现改为 0 < n <= 5
        # 使得 Eagle3 验证（batch size 5）也能使用 wvSplitK 内核
        if m > 8 and 0 < n <= 5:
            cu_count = num_compute_units()
            out = ops.wvSplitK(weight, x_view, cu_count, bias)
            return out.reshape(*x.shape[:-1], weight.shape[0])
        elif m % 4 == 0 and n == 1 and k <= 8192 and bias is None:
            out = ops.LLMM1(weight, x_view, 4)
            return out.reshape(*x.shape[:-1], weight.shape[0])

    # fallback 到其他内核或 torch.nn.functional.linear
```

csrc/rocm/skinny_gemms.cu

添加了 N=5 的 tile 配置，使内核能处理新的 batch size。

```
// csrc/rocm/skinny_gemms.cu (wvSplitK 函数中新增 case 5)
torch::Tensor wvSplitK(const at::Tensor& in_a, const at::Tensor& in_b, ...) {
    // ... 前面的 switch 处理 N=1..4 ...
    case 5:
        // 新增 N=5 的 tile 配置
        // use_wave32 为 true 时使用 32x16 tile
        // 否则使用 64x16 tile
        if (use_wave32)
            WVSPLIT_TILE_CFG(32, 16, sYT, 5)
        else
            WVSPLIT_TILE_CFG(64, 16, sYT, 5)
        break;
    default:
        throw std::runtime_error(
            "Unsupported N value: " + std::to_string(M_in) + "," + ...);
    // ... 后续处理 ...
}
```

评论区精华

审核者 tjtanaa 询问此改动是否仅针对 N=5 有提升，以及能否进一步增加 N 的范围（如 N<16）。作者回应称，增加 N 的特殊化会显著增加编译时间（因涉及其他模板参数），目前未验证更高 N 的性能。另外 tjtanaa 提到 torch.nn.linear 被包装在 custom op 中，未受 torch.compile 优化，需要考虑 custom ops 的优化。

- 是否可进一步扩展 N 支持范围 (question): 当前仅支持到 N=5，未来如需更大 N 需权衡编译时间与收益。

风险与影响

- 风险：风险较低。仅修改了 dispatch 条件（n<=5）和添加了一个 switch case，不影响已有功能。主要风险在于 kernel 编译时间可能因新增 specialization 而略有增加，但已在讨论中确认可接受。此外，由于未对 N>5 的场景进行优化，若未来有更大 batch 的验证需求，仍需 fallback。
- 影响：直接影响 ROCm 平台上使用推测解码（Eagle3 等）的 Qwen3-8B 等模型，验证阶段 batch size 为 5 时可获得 12-14% 的性能提升。对其他模型或非推测解码场景无影响。改动仅 7 行代码，无配置或接口变更，无兼容性问题。
- 风险标记：仅限 ROCm 平台，增加编译时间

关联脉络

- 暂无明显关联 PR