PR #46474 完整报告

vllm-project/vllm

[ROCm][Perf] Fused shared expert for Minimax M3

合并时间：2026-06-27 20:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46474

执行摘要

- 一句话：融合共享专家至 AITER MoE，提升 MiniMax-M3 TPOT 4-17%
- 推荐动作：此 PR 值得精读，特别是它展示了如何在不破坏现有非 AITER 路径的前提下，引入 AITER 专属的优化路径。设计上采用多函数共存、条件分支清晰，是平台特定性能优化的良好案例。建议工程师关注 _aiter_moe_fused_shared_experts_enabled 的设计以及如何利用 aiter 的 grouped top-k 实现共享专家融合。如果后续需要为其他模型做类似优化，可以参考此模式。

功能与动机

MiniMax-M3 模型包含一个始终激活的共享专家，原有实现将其作为独立的密集 MLP 运行，增加了 kernel 启动和显存访问。融合共享专家到 routed MoE 的一次调用中，可以显著减少延迟。本 PR 是 #46419（AITER MoE 后端选择）的后续，进一步利用 AITER 的 grouped top-k 能力来支持融合共享专家，从而最大化性能。

实现拆解

实现分以下几步：

1. 新增 AITER 专用融合检测函数：在 vllm/models/minimax_m3/amd/model.py 中添加 _aiter_moe_fused_shared_experts_enabled()，判断是否满足：已启用 FSE（环境变量 VLLM_ROCM_USE_AITER_FUSION_SHARED_EXPERTS）、不在专家并行下、运行在 gfx950 且 AITER 的 is_fusion_moe_shared_experts_enabled() 返回 True。该函数专用于 AITER MoE 路径，与 #46545 已有的 fse_topk_bias_router_enabled（适用于非 AITER）共存。
2. 改造 MiniMaxM3MoE 初始化：在 MiniMaxM3MoE.__init__ 中，根据 self.use_aiter_moe_fse（源自上述函数）设置不同的 gating 参数：当使用 AITER 路径时，启用 use_grouped_topk=True、num_expert_group=1、topk_group=1 以通过 aiter 的 biased grouped top-k；同时 apply_routed_scale_to_output 设为 False（aiter 内部处理缩放）。非 AITER 路径保持原有 vLLM topk router 行为。
3. 调整量子化 MoE 后端（vllm/model_executor/layers/quantization/quark/quark_moe.py 及相关文件）：为支持新路径中的中间维度填充需求，在 FusedMoEConfig 中添加 intermediate_size_per_partition_unpadded 和 hidden_dim_unpadded 等配置项，允许模型指定未填充的维度，避免因填充导致 kernel 计算错误。

4. 性能验证：通过 GSM8K 评估（5-shot）验证准确率无回归，并在 ISL/OSL 8K1K 场景下测试 TPOT，得到 4.5%-16.9% 的加速。

注意：此 PR 依赖 #46419 的 AITER MoE 后端和 #46545 的 FSE Router，三者共同提供全路径优化。

关键文件：

- vllm/models/minimax_m3/amd/model.py（模块 模型；类别 source；类型 core-logic；符号 _fuse_shared_experts_enabled, _aiter_moe_fused_shared_experts_enabled, MiniMaxM3MoE.init, MiniMaxM3MoE.forward）：核心变更文件，新增 AITER 专用融合检测函数、改造 MiniMaxM3MoE 条件分支以实现共享专家融合。

关键符号：_fuse_shared_experts_enabled, _aiter_moe_fused_shared_experts_enabled, _m3_fuse_shared_experts_enabled, MiniMaxM3MoE.init, MiniMaxM3MoE.forward, MiniMaxM3MoE.load_weights

关键源码片段

vllm/models/minimax_m3/amd/model.py

核心变更文件，新增 AITER 专用融合检测函数、改造 MiniMaxM3MoE 条件分支以实现共享专家融合。

```
# vllm/models/minimax_m3/amd/model.py

def _aiter_moe_fused_shared_experts_enabled(config: PretrainedConfig) -> bool:
    """判断是否通过AITER的biased grouped top-k实现共享专家融合。

    条件：必须已启用FSE（`_fuse_shared_experts_enabled`返回True）、
    非专家并行、运行在gfx950且AITER内部表示支持融合。
    """
    if not _fuse_shared_experts_enabled(config):
        return False
    from vllm.platforms.rocm import on_gfx950
    # 此时已确认是 ROCM 平台，import 安全
    return on_gfx950() and rocm_aiter_ops.is_fusion_moe_shared_experts_enabled()


class MiniMaxM3MoE(nn.Module):
    def __init__(
        self,
        config: PretrainedConfig,
        prefix: str = "",
    ):
        super().__init__()
        # … 其他初始化 …

        # 判断是否使用 AITER 专用 FSE 路径（条件：AITER MOE 激活且 gfx950）
        self.use_aiter_moe_fse = _m3_fuse_shared_experts_enabled(config)
```

```
# 当使用 AITER FSE 时，设定 grouped top-k 参数为单组（退化为普通 top-k 但走 aiter
biased 路线）
# 同时 `apply_routed_scale_to_output` 取反（aiter 内部处理缩放）
self.gate = GroupedTopKRouting(
    self.num_total_experts,
    self.top_k,
    scoring_func=config.scoring_func,
    e_score_correction_bias=self.e_score_correction_bias,
    renormalize=True,
    use_grouped_topk=self.use_aiter_moe_fse,
    num_expert_group=1 if self.use_aiter_moe_fse else None,
    topk_group=1 if self.use_aiter_moe_fse else None,
    activation="swigluoai_uninterleave",
    swiglu_limit=config.swiglu_limit,
    swiglu_alpha=config.swiglu_alpha,
    apply_routed_scale_to_output=not self.use_aiter_moe_fse,
)
# 其他初始化 ...
```

评论区精华

Review 中主要讨论了以下要点：

- AITER 导入位置：tjtanaa 建议将 `from aiter.ops.flydsl.moe_common import GateMode` 移到函数内部，避免在没有 aiter 的环境下造成导入错误。Fangzhou-Ai 采纳，将其置于特定条件分支内。
- `activation_interleave` 未定义风险：dllehr-amd 指出 `activation_interleave` 变量只在 `swigluoai` 等特定激活下定义，在其他激活（如 gelu）下未定义。要求设默认值 None，并清理 `GateMode.INTERLEAVE` 的误用。作者修复。
- 两个 FSE 函数共存：tjtanaa 要求保留 #46545 中的 `fse_topk_bias_router_enabled` 作为非 AITER 路径的入口，新增 `_aiter_moe_fused_shared_experts_enabled` 仅控制 AITER 路径，并通过 `_m3_fuse_shared_experts_enabled` 作为 OR 合并。作者执行。
- CK MoE 后端选择机制：BowenBao 批评直接在 `quark_moe.py` 中添加 `_ck_moe_enabled` 判断不符合 oracle/kernel backend 设计理念，应通过 `oracle` 统一调度。Fangzhou-Ai 后续重构了该部分，最终获得批准。
- `load_weights` 修改必要性：tjtanaa 质疑为何修改 weight 加载逻辑，认为 mxfp4 和 mxfp8 权重参数名一致。Fangzhou-Ai 验证后 revert 了该改动。
  - AITER 导入位置安全性 (design): Fangzhou-Ai 将导入放入条件分支内，确保只在需要时加载。
  - activation_interleave 默认值未定义风险 (correctness): 作者给 activation_interleave 添加默认值 None，并清理了相关的 GateMode.INTERLEAVE 误用。
  - 保留两个 FSE 函数分别对应 aiter 和非 aiter (design): 作者保留 fse_topk_bias_router_enabled（重命名自 _fuse_shared_experts_enabled），新增 _aiter_moe_fused_shared_experts_enabled，并使用 _m3_fuse_shared_experts_enabled 作为 OR 组合。

- CK MoE 后端选择不应绕过 oracle 机制 (design): Fangzhou-Ai 移除了该独立判断，改为在 oracle 中进行后端选择？最终 BowenBao APPROVED，可能通过其他方式解决。
- load_weights 修改不必要 (correctness): Fangzhou-Ai 验证后 revert 了该修改，保持与 #46545 一致。

风险与影响

- 风险：
  - 核心路径变更：MoE 是模型前向的关键部分，引入分支可能影响非 AITER 路径的稳定性。需确保 use_grouped_topk 等参数在非 AITER 下不变。
  - 平台特定：代码假设 AITER 仅存在于 ROCM/gfx950，在其他平台上不会触发，但若未来有其他平台支持 AITER，需要调整守卫条件。
  - 依赖绑定：该优化依赖 aiter 库的最低版本（v0.1.15.post3），若用户使用旧版 aiter，可能导致功能不可用或错误。PR body 中作者明确要求等待 aiter bump 合并后才可合并（但最终未体现，可能需要后续跟进）。
  - 测试覆盖：没有看到直接针对融合共享专家的测试新增，依赖手工验证（GSM8K 和 perf sweep）。未来若其他模型复用此机制，需要更好的测试。
  - 配置理解成本：新增多个环境变量（如 VLLM_ROCM_USE_AITER_FUSION_SHARED_EXPERTS），用户可能不清楚何时启用，可能造成预期外的行为。
- 影响：
  - 用户影响：使用 AMD ROCm gfx950 并搭载 MiniMax-M3 模型的用户可获得 4.5%-17% 的 TPOT 性能提升，但需设置环境变量 VLLM_ROCM_USE_AITER_FUSION_SHARED_EXPERTS=1 并确保 aiter 版本满足要求。无需额外的代码改动即可享受加速。
  - 团队影响：维护成本上升，因为现在有两个 FSE 路径（vLLM router 和 AITER），需要保持两者行为一致且不退化。未来模型若想复用此机制，需类似的条件分支设计。
  - 系统影响：对非 ROCM 平台无影响，对 gfx950 以外的 GPU 无影响。
  - 风险标记：核心路径变更，平台特定依赖，环境变量控制，缺少测试覆盖，依赖 aiter 版本

关联脉络

- PR #46419 [ROCm][Perf] Use flydsl moe with Minimax-M3 mxfp8 weights on gfx950 and implemented moe-backend selection: 本 PR 是 #46419 的 follow-up，在其 AITER MoE 后端基础上融合共享专家。
- PR #46545 [ROCm][Perf] Fused shared expert topk bias router: 引入了 vLLM router 的共享专家融合（fuse_topk_bias_router），本 PR 在其基础上新增 AITER 专用路径。