PR #31889 完整报告

sgl-project/sglang

[AMD] Cache AITER expert mask across decode

合并时间：2026-07-22 22:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31889

执行摘要

- 一句话：AMD AITER MoE 专家掩码缓存优化
- 推荐动作：推荐合并。该 PR 是典型的微小性能优化，改动清晰且经过内部验证，无副作用。Approver 已批准，可直接合并。

功能与动机

AITER MoE runner 在每个 decode 步骤都从固定的本地专家映射重建 expert_mask_gpu，包含不必要的 host→device 拷贝。PR body 明确说明 'recomputing the mask (which includes a host→device copy) on every decode step is wasteful'。

实现拆解

1. 添加缓存守卫：在 python/sglang/srt/layers/moe/token_dispatcher/standard.py 的 dispatch 方法中，将 AITER 分支条件从 if self.use_aiter_moe_runner 修改为 if self.use_aiter_moe_runner and self.expert_mask_gpu is None，使得 expert_mask_gpu 仅在第一次调用时计算并赋值，后续 decode 步骤直接复用已缓存的张量。
2. 调整分支逻辑：将原 else 分支改为 elif not self.use_aiter_moe_runner，确保非 AITER 路径的专家映射逻辑不受影响，保持行为一致。
3. 无需额外状态初始化：expert_mask_gpu 是 StandardDispatcher 的现有成员，默认值为 None，因此修改不涉及构造函数改动。

关键文件：

- python/sglang/srt/layers/moe/token_dispatcher/standard.py（模块 MoE 调度；类别 source；类型 core-logic）：核心变更文件，通过修改条件守卫避免 AITER expert_mask_gpu 重复计算

关键符号：dispatch

关键源码片段

python/sglang/srt/layers/moe/token_dispatcher/standard.py

核心变更文件，通过修改条件守卫避免 AITER expert_mask_gpu 重复计算

```
# python/sglang/srt/layers/moe/token_dispatcher/standard.py
```

```
class StandardDispatcher:
    # ... 其他逻辑 ...

    def dispatch(self, hidden_states, topk_output, hidden_states_scale=None):
        # ... 前置处理 ...

        if self.local_expert_mapping is not None and not self.skip_local_expert_mapping:
            # 仅当 expert_mask_gpu 尚未初始化时才构建，后续 decode 步骤直接缓存命中
            if self.use_aiter_moe_runner and self.expert_mask_gpu is None:
                # 构建布尔掩码并转换为 int32 后拷贝到 CUDA 设备，只发生一次
                self.expert_mask_gpu = (
                    (
                        (self.local_expert_mapping >= 0)
                        & (self.local_expert_mapping < self.num_local_experts)
                    )
                    .to(torch.int32)
                    .to(device="cuda")
                )
            elif not self.use_aiter_moe_runner:
                # 非 AITER 路径保持不变，对 topk_ids 进行专家映射
                if TopKOutputChecker.format_is_standard(topk_output):
                    topk_output = topk_output._replace(
                        topk_ids=self.local_expert_mapping[topk_output.topk_ids]
                    )
                elif TopKOutputChecker.format_is_triton_kernels(topk_output):
                    raise NotImplementedError()

        # ... 返回 DispatchOutput...
```

评论区精华

该 PR 没有 review 评论讨论。仅有 Approver HaiShaw 批准，未出现争议或设计权衡讨论。

- 暂无高价值评论线程

风险与影响

- 风险：变更仅涉及一个条件判断，逻辑简单，风险极低。潜在风险：若 expert_mask_gpu 在后续 decode 间被外部修改（无此类代码），缓存可能导致过时掩码；但当前设计中该属性仅在此处写入，且配置为 is None 守卫，安全。
- 影响：影响范围极小：仅影响启用 AITER MoE runner 的 AMD 平台。每次 decode 步骤减少一次 host→device 拷贝和掩码重建，性能提升随总 decode 步骤数线性累积。非 AMD 或未使用 AITER 的场景零影响。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR