PR #26746 完整报告

sgl-project/sglang

Support optional kwargs in AITER fused_moe runner

合并时间：2026-06-04 23:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26746

执行摘要

- 一句话：为 AITER fused_moe 添加可选 kwargs 和 no_combine 支持
- 推荐动作：建议快速合并。PR 设计清晰，测试全面。关键设计决策（functools.cache 特征探测、条件 kwargs 转发、空输入适配）值得其他 runner 参考。

功能与动机

根据 PR 描述，一些 AITER fused_moe 集成需要将后端特定选项直接传递给已安装的 AITER 内核，同时保持通用 SGLang MoE runner 接口不变。此 PR 添加了一个小的扩展点用于那些可选 kwargs，并且在支持时启用 no_combine 转发。

实现拆解

1. 在 AiterMoeQuantInfo 中添加 fused_moe_kwargs: Optional[dict[str, Any]] 字段，允许调用方传入任意额外参数。
2. 新增 _aiter_fused_moe_supports_no_combine 函数，使用 functools.cache 和 inspect.signature 探测安装的 aiter.fused_moe 是否接受 no_combine。
3. 修改 AiterRunnerCore.run：将原来的 assert 替换为条件检查，若配置 no_combine=True 但底层不支持则抛出 NotImplementedError；若支持则将 no_combine=True 加入 extra 字典。
4. 调整空输入处理：当输入为空且 no_combine=True 时，返回 (0, topk, hidden) 形状的空张量。
5. 新增 test/registered/unit/layers/moe/test_aiter_runner.py，包含三个 CPU 单元测试：验证 kwargs 转发、不支持时的异常、空输入形状保持。

关键文件：

- python/sglang/srt/layers/moe/moe_runner/aiter.py（模块 MoE 运行器；类别 source；类型 dependency-wiring；符号 _aiter_fused_moe_supports_no_combine, AiterMoeQuantInfo.fused_moe_kwargs, AiterRunnerCore.run）：核心修改文件，新增 fused_moe_kwargs 字段和 no_combine 支持
- test/registered/unit/layers/moe/test_aiter_runner.py（模块 单元测试；类别 test；类型 test-coverage；符号 _runner_input, _quant_info, _install_fake_aiter, test_aiter_runner_forwards_no_combine_and_extra_fused_moe_kwargs）：新增测试文件，覆盖三个关键场景

关键符号：_aiter_fused_moe_supports_no_combine, AiterRunnerCore.run

关键源码片段

python/sglang/srt/layers/moe/moe_runner/aiter.py

核心修改文件，新增 fused_moe_kwargs 字段和 no_combine 支持

```
import functools
import inspect

@functools.cache
def _aiter_fused_moe_supports_no_combine() -> bool:
    """探测安装的 aiter.fused_moe 是否接受 no_combine 参数。"""
    from aiter.fused_moe import fused_moe
    return "no_combine" in inspect.signature(fused_moe).parameters

class AiterRunnerCore(MoeRunnerCore):
    def run(self, runner_input, quant_info, running_state, hooks=None):
        # 检查 no_combine 兼容性
        if self.config.no_combine and not _aiter_fused_moe_supports_no_combine():
            raise NotImplementedError(
                "no_combine=True requested but the installed aiter.fused_moe does "
                "not accept a `no_combine` kwarg. ..."
            )

        # 空输入处理
        if runner_input.hidden_states.shape[0] == 0:
            if self.config.no_combine:
                # 保持输出秩 (0, topk, hidden)
                topk = runner_input.topk_ids.shape[-1]
                hidden_size = runner_input.hidden_states.shape[-1]
                return AiterRunnerOutput(
                    hidden_states=runner_input.hidden_states.new_empty((0, topk, hidden_size))
                )
            return AiterRunnerOutput(hidden_states=runner_input.hidden_states)

        # 构建 extra 字典
        extra: dict = {}
        if quant_info.fused_moe_kwargs:
            extra.update(quant_info.fused_moe_kwargs)
        # 其他条件参数 ...
        if self.config.no_combine:
            extra["no_combine"] = True

        output = fused_moe(..., **extra)
```

评论区精华

PR 主要讨论集中在 CI 失败上：新增测试文件缺少 if __name__== "__main__": 入口导致 run_suite.py 硬失败。HaiShaw 在第三次提交中修复，CI 通过。此外通过 /tag-and-rerun-ci 触发额外 CI。无技术争议。

- CI 失败：测试文件缺少 main__入口 (testing): 添加了 if__name== "main": 入口，CI 恢复。

风险与影响

- 风险：风险较低。新字段默认 None，no_combine 默认 False，不影响现有路径。若用户启用 no_combine 但 AITER 版本不支持，会得到清晰的 NotImplementedError，不会静默失败。单元测试覆盖所有分支（含空输入）。兼容性通过特征探测保障。
- 影响：影响范围局限在 AITER runner 的 MoE 推理。对非 AITER 用户无影响。AMD 用户可开始利用 no_combine 特性（需 AITER 版本支持）。此模式可复用于其他后端。
- 风险标记：新功能默认不启用，不影响现有逻辑

关联脉络

- 暂无明显关联 PR