PR #29270 完整报告

sgl-project/sglang

[sgl-kernel/cpu]: fix arm64 w8a8 moe kernel signature

合并时间：2026-06-26 08:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29270

执行摘要

- 一句话：修复 arm64 MoE 内核函数签名不一致
- 推荐动作：该 PR 为典型的跨平台签名同步修复，变更简洁且必要。开发者可以快速了解 ARM64 平台如何维护函数签名一致性。值得关注的是作者采用的方式是将公共声明统一化，而非为 ARM64 保留特殊路径，这种设计值得借鉴。

功能与动机

fused_experts_cpu 函数签名已变更，ARM64 实现需同步更新。PR 描述明确提到 "fused_experts_cpu signature has changed, arm64 implementation needs update."

实现拆解

1. 统一函数声明：在 sgl-kernel/csrc/cpu/torch_extension_cpu.cpp 中，将 fused_experts_cpu 的声明从原先 ARM64 专用的 #else 分支中移出，放在公共区域，并加入新增的 w1_bias、w2_bias、alpha、limit 四个参数，确保跨平台签名一致。
2. 更新 ARM64 实现：在 sgl-kernel/csrc/cpu/aarch64/moe.cpp 中，为 fused_experts_cpu 函数增加四个形参（均添加 /* */ 注释以表示暂未使用），使函数签名与声明匹配。
3. 删除 ARM64 专用重载：移除之前 ARM64 平台上 #else 分支中的简化版 fused_experts_cpu 声明，消除重复定义和潜在冲突。
4. 纠正 TORCH_LIBRARY 注册：将 fused_experts_cpu 的 m.def / m.impl 从 #if ! defined(...) 条件块中移出，确保在 ARM64 平台上也能正确注册新签名。同时删除 ARM64 专属的简化注册分支。
5. 更新测试调用：修改 test/registered/cpu/arm64/test_moe.py 中的 _int8_moe 函数，在调用 kernel.fused_experts_cpu 时补上四个 None 参数，以匹配新签名。

关键文件：

- sgl-kernel/csrc/cpu/torch_extension_cpu.cpp（模块 内核调度；类别 source；类型 core-logic；符号 fused_experts_cpu）：核心调度文件，调整了 fused_experts_cpu 的声明位置和条件编译逻辑，移除了 ARM64 专用后门，统一所有平台签名。
- sgl-kernel/csrc/cpu/aarch64/moe.cpp（模块 ARM64 内核；类别 source；类型 core-logic；符号 fused_experts_cpu）：ARM64 平台 MoE 内核实现，新增四个形参以匹配更新的签名。参数名以注释形式标注以避免编译警告。

- test/registered/cpu/arm64/test_moe.py（模块 测试；类别 test；类型 test-coverage；符号 _int8_moe）：ARM64 MoE 测试用例，更新调用签名增加四个 None 参数以匹配新函数签名。

关键符号：fused_experts_cpu

关键源码片段

sgl-kernel/csrc/cpu/torch_extension_cpu.cpp

核心调度文件，调整了 fused_experts_cpu 的声明位置和条件编译逻辑，移除了 ARM64 专用后门，统一所有平台签名。

```
// 移除了原先 #if !defined(SGLANG_CPU_ARM64_SKIP_X86_ONLY_OPS) 对 fused_experts_cpu
声明的包裹
// 并将声明放在公共区域，确保所有平台共享同一签名
// 结构变化：删除了 ARM64 专用的简化版本，统一使用完整参数版本

// 公共区域声明（不再被条件编译分隔）
at::Tensor fused_experts_cpu(
    at::Tensor& hidden_states,
    at::Tensor& w1,
    at::Tensor& w2,
    at::Tensor& topk_weights,
    at::Tensor& topk_ids,
    bool inplace,
    int64_t moe_comp_method,
    const std::optional<at::Tensor>& w1_scale,
    const std::optional<at::Tensor>& w2_scale,
    const std::optional<at::Tensor>& w1_zero,
    const std::optional<at::Tensor>& w2_zero,
    const std::optional<std::vector<int64_t>> block_size,
    const std::optional<at::Tensor>& w1_bias, // 新增参数
    const std::optional<at::Tensor>& w2_bias, // 新增参数
    const std::optional<double>& alpha, // 新增参数
    const std::optional<double>& limit, // 新增参数
    bool is_vnni);

// TORCH_LIBRARY 注册同样移到公共区域，确保 ARM64 也能注册完整签名
m.def(
    "fused_experts_cpu(Tensor hidden_states, Tensor w1, Tensor w2, Tensor topk_weights,
    Tensor topk_ids, bool "
    "inplace, int moe_comp_method, Tensor? w1_scale, Tensor? w2_scale, "
    "Tensor? w1_zero, Tensor? w2_zero, int[]? block_size, Tensor? w1_bias, Tensor? w2_bias,
    float? alpha, float? "
    "limit, bool is_vnni) -> Tensor");
m.impl("fused_experts_cpu", torch::kCPU, &fused_experts_cpu);
```

sgl-kernel/csrc/cpu/aarch64/moe.cpp

ARM64 平台 MoE 内核实现，新增四个形参以匹配更新的签名。参数名以注释形式标注以避免编译警告。

```
// ARM64 实现的 fused_experts_cpu 函数签名更新：增加四个形参以匹配公共声明
// 当前实现中这些参数被标记为 `/* unused */`，避免编译警告
at::Tensor fused_experts_cpu(
    at::Tensor& hidden_states,
    at::Tensor& w13,
    at::Tensor& w2,
    at::Tensor& topk_weights,
    at::Tensor& topk_ids,
    bool inplace,
    int64_t moe_comp_method,
    const std::optional<at::Tensor>& w13_scale,
    const std::optional<at::Tensor>& w2_scale,
    const std::optional<at::Tensor>& /*w13_zero*/,
    const std::optional<at::Tensor>& /*w2_zero*/,
    const std::optional<std::vector<int64_t>> block_size,
    const std::optional<at::Tensor>& /*w1_bias*/, // 新增
    const std::optional<at::Tensor>& /*w2_bias*/, // 新增
    const std::optional<double>& /*alpha*/, // 新增
    const std::optional<double>& /*limit*/, // 新增
    bool /*is_vnni*/) {
  // ... 函数体保持不变，仅签名更新
  const auto st = hidden_states.scalar_type();
  CHECK_INPUT(hidden_states);
  // ...
}
```

test/registered/cpu/arm64/test_moe.py

ARM64 MoE 测试用例，更新调用签名增加四个 None 参数以匹配新函数签名。

```
# 调用 fused_experts_cpu 时补充四个 None 参数以匹配新签名
out = kernel.fused_experts_cpu(
    a,
    packed_w1,
    packed_w2,
    topk_weight,
    topk_ids.to(torch.int32),
    inplace,
    CPUQuantMethod.INT8_W8A8,
    w1_s,
    w2_s,
    None,
    None,
    None,
    None, # w1_bias
    None, # w2_bias
    None, # alpha
```

```
        None, # limit
        prepack,
    )
```

评论区精华

作者在 review 评论中明确了两个变更点的意图：

1) moe.cpp 中添加四个参数是为了匹配最新的 fused_experts_cpu 签名； 2) 删除 ARM64 专用后门代码是因为 fused_experts_cpu 应在所有平台上共享同一原型。没有产生争议。

- ARM64 实现增加参数以匹配签名 (correctness): 确认一致，无争议。
- 删除 ARM64 专用后门代码 (design): 接受该设计决策。

风险与影响

- 风险：变更集中在 ARM64 平台类型签名和声明流程调整，不涉及核心算法逻辑改动，回归风险低。但测试仅覆盖 INT8 W8A8 量化方式，若后期 ARM64 支持更多量化组合，测试可能不足。
- 影响：直接影响 ARM64 平台上 w8a8 MoE 内核的编译和运行，修复了因函数签名不一致导致的链接失败问题。其他平台无影响。对用户透明，但确保 ARM64 用户能正常使用 MoE 功能。
- 风险标记：ARM64 专有变更，接口签名变更

关联脉络

- 暂无明显关联 PR