PR #31110 完整报告

sgl-project/sglang

[CPU] bypass scoring_func argument in topk for cpu device

合并时间：2026-07-14 21:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31110

执行摘要

- 一句话：修复 CPU 设备 topk 接口不兼容导致 CI 失败
- 推荐动作：该 PR 是快速的兼容性修复，适合合入以修复 CI。建议关注后续对 test_spec_eagle_parity_cpu 的修复 PR。

功能与动机

修复 CI 测试 test_latency_fp8_moe_model 的失败，根本原因是 fused_topk_cpu 函数签名未同步更新，导致参数不匹配。PR body 中明确引用了该失败的 Traceback。

实现拆解

1. 更新 fused_topk_cpu 函数签名：在 python/sglang/srt/layers/moe/topk.py 中新增 routed_scaling_factor、apply_routed_scaling_factor_on_output、num_fused_shared_experts、packed_out、num_token_non_padded 参数，并添加参数校验（若使用不受支持的特性则抛出 ValueError）。
2. 更新 grouped_topk_cpu 函数签名：在 python/sglang/srt/layers/moe/topk.py 中新增 scoring_func 参数，并在非 softmax 时抛出 ValueError。
3. 临时禁用 CPU 上的 EAGLE3 精度测试：在 test/registered/cpu/test_spec_eagle_parity_cpu.py 中通过 register_cpu_ci 的 disabled 参数禁用 test_spec_eagle_parity_cpu 测试，待后续修复数值精度问题。

关键文件：

- python/sglang/srt/layers/moe/topk.py（模块 MoE；类别 source；类型 core-logic；符号 fused_topk_cpu, grouped_topk_cpu）：核心源码文件，修改了 CPU 设备上 topk 函数的接口，添加参数校验和新的可选参数，是修复 CI 失败的关键。
- test/registered/cpu/test_spec_eagle_parity_cpu.py（模块 推测解码；类别 test；类型 test-coverage）：测试文件，通过注册 CI 时使用 disabled 参数暂时禁用 EAGLE3 CPU 精度测试，避免因已知的数值精度问题持续失败。

关键符号：fused_topk_cpu, grouped_topk_cpu

关键源码片段

python/sglang/srt/layers/moe/topk.py

核心源码文件，修改了 CPU 设备上 topk 函数的接口，添加参数校验和新的可选参数，是修复 CI 失败的关键。

```
# python/sglang/srt/layers/moe/topk.py

def fused_topk_cpu(
    hidden_states: torch.Tensor,
    gating_output: torch.Tensor,
    topk: int,
    renormalize: bool,
    correction_bias: torch.Tensor = None,
    scoring_func: str = "softmax",
    # 以下为新增参数，用于兼容统一接口但 CPU kernel 尚不支持
    routed_scaling_factor: Optional[float] = None,
    apply_routed_scaling_factor_on_output: Optional[bool] = False,
    num_fused_shared_experts: int = 0,
    packed_out: Optional[torch.Tensor] = None,
    num_token_non_padded: Optional[torch.Tensor] = None,
):
    # 参数校验：CPU 版本不支持共享 expert、routed scaling 等特性
    if num_fused_shared_experts != 0:
        raise ValueError(
            f"num_fused_shared_experts must be 0 for CPU fused topk, got: {num_fused_shared_
            experts}"
        )
    if apply_routed_scaling_factor_on_output:
        raise ValueError(
            "apply_routed_scaling_factor_on_output is not supported for CPU fused topk"
        )
    if packed_out is not None:
        raise ValueError("packed_out is not supported for CPU fused topk")
    if num_token_non_padded is not None:
        raise ValueError("num_token_non_padded is not supported for CPU fused topk")

    # fallback 到 torch native 实现（支持非 softmax scoring 或 correction_bias）
    if correction_bias is not None or scoring_func != "softmax":
        return fused_topk_torch_native(
            hidden_states, gating_output, topk, renormalize,
            correction_bias=correction_bias, scoring_func=scoring_func,
        )

    # 默认走 CPU kernel
    topk_weights, topk_ids = torch.ops.sgl_kernel.topk_softmax_cpu(
        hidden_states=hidden_states, gating_output=gating_output,
        topk=topk, renormalize=renormalize,
    )
    return topk_weights, topk_ids

def grouped_topk_cpu(
    hidden_states: torch.Tensor,
```

```
    gating_output: torch.Tensor,
    topk: int,
    renormalize: bool,
    num_expert_group: Optional[int] = None,
    topk_group: Optional[int] = None,
    num_fused_shared_experts: int = 0,
    routed_scaling_factor: Optional[float] = None,
    apply_routed_scaling_factor_on_output: Optional[bool] = False,
    scoring_func: str = "softmax", # 新增参数
):
    assert not apply_routed_scaling_factor_on_output
    # 仅支持 softmax scoring
    if scoring_func != "softmax":
        raise ValueError(f"Unsupported scoring function: {scoring_func}")

    return torch.ops.sgl_kernel.grouped_topk_cpu(
        hidden_states, gating_output, topk, renormalize,
        num_expert_group, topk_group, num_fused_shared_experts,
        routed_scaling_factor,
        num_token_non_padded=None,
    )
```

test/registered/cpu/test_spec_eagle_parity_cpu.py

测试文件，通过注册 CI 时使用 disabled 参数暂时禁用 EAGLE3 CPU 精度测试，避免因已知的数值精度问题持续失败。

```
# test/registered/cpu/test_spec_eagle_parity_cpu.py

register_cpu_ci(
    est_time=480,
    suite="base-b-test-cpu",
    # 临时禁用：EAGLE3 在 CPU intel_amx 上存在数值精度不匹配，等待后续修复
    disabled="EAGLE3 numerical parity mismatches on CPU intel_amx",
)
```

评论区精华

PR 提交者 mingfeima 在 Issue 评论中要求 @htzo 先禁用 test_spec_eagle_parity_cpu，并指出后续需要修复该问题。其他 review 讨论未提供更多细节。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 回归风险：新增参数校验可能使部分依赖非标准 scoring_func 的 CPU 流水线触发 ValueError，但当前代码已通过 fallback 到 fused_topk_torch_native 来兼容非 softmax 场景，风险可控。

2. 测试覆盖缺失：临时禁用的 EAGLE3 精度测试长期未修复可能掩盖回归，需要持续跟踪。

- 影响：
  1. 用户影响：仅影响 CPU 设备（Intel AMX）上的 MoE topk 逻辑，修复后 CI 可恢复通过，其他设备无影响。
  2. 系统影响：限制 CPU 设备上部分 topk 参数的使用（如 scoring_func 仅为 softmax 受支持），但符合预期行为。
  3. 团队影响：需要后续修复 EAGLE3 CPU 精度问题后重新启用测试。 - 风险标记：测试被禁用，临时绕过未完全修复

关联脉络

- 暂无明显关联 PR