PR #35453 完整报告

sgl-project/sglang

[Fix] Support LSE on the RadixAttention extra-kwargs graph path

合并时间：2026-08-29 09:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35453

执行摘要

- 一句话：修复 extra-kwargs 注意力路径丢弃 LSE 输出
- 推荐动作：值得关注，因为修复了一个图形路径下的功能缺陷，并提供了清晰的测试验证。设计决策上，通过检测 return_lse 键和 mha_return_lse 标志来决定是否处理 LSE，保持了向后兼容性。

功能与动机

在 PR body 中，作者指出："RadixAttention routes backend-specific arguments such as relative bias and auxiliary tensors through a separate graph-compatible helper. Unlike the standard attention path, that helper assumes a tensor-only backend result and discards log-sum-exp output. As a result, attention calls that require both extra kwargs and LSE cannot use the graph path correctly." 即当注意力调用同时需要额外参数（如相对偏置）和 LSE 时，无法正确使用图路径。

实现拆解

实现分为三个步骤：

1. 扩展 kwargs 检测列表：在 RadixAttention.forward 中，将 return_lse 添加到检测额外参数的列表中，使得该路径能处理 LSE 请求。
2. 修改 attention_with_output_extra_kwargs 函数：将返回类型从 None 改为 Optional[torch.Tensor]，并在内部根据 return_lse 标志解析后端返回的 (output, lse) 元组或纯 tensor，保留原 tensor-only 行为。
3. 在调用处处理 LSE：forward 中现在接收 lse 变量，并在 return_lse 时返回 (output.view(-1, ...), lse)；同时在 attention_with_output_extra_kwargs 内部，若 LSE 形状与 bucket 不一致，则创建零填充的 padded_lse 以保证 CUDA 图重放时的稳定形状。
4. 测试配套：新增 test_extra_kwargs_path_returns_bucket_shaped_lse 验证真实 token 的 LSE 值和填充零行。

关键文件：

- python/sglang/srt/layers/radix_attention.py（模块 注意力层；类别 source；类型 core-logic）：核心修改文件，修复了 extra-kwargs 路径丢弃 LSE 的问题。

- test/registered/unit/layers/test_radix_attention.py（模块 注意力层；类别 test；类型 test-coverage；符号 test_extra_kwargs_path_returns_bucket_shaped_lse）：新增针对 extra-kwargs 路径的 LSE 回归测试。

关键符号：attention_with_output_extra_kwargs, RadixAttention.forward

关键源码片段

python/sglang/srt/layers/radix_attention.py

核心修改文件，修复了 extra-kwargs 路径丢弃 LSE 的问题。

```
# python/sglang/srt/layers/radix_attention.py
def attention_with_output_extra_kwargs(...) -> Optional[torch.Tensor]:
    """处理特殊参数（如 aux_tensors）的注意力路径，现在支持返回 LSE。"""
    # ... 前置逻辑（调用后端）
    forward_batch.out_cache_loc = original_out_cache_loc
    # 判断是否需要返回 LSE：来自 kwargs 或 forward_batch
    return_lse = bool(kwargs.get("return_lse") or forward_batch.mha_return_lse)
    if return_lse:
        # 后端返回 (output, lse) 元组
        assert isinstance(ret, tuple)
        ret, lse, *_ = ret
    else:
        # 保持 tensor-only 行为
        assert isinstance(ret, torch.Tensor)
        lse = None
    # 拷贝实际 token 输出到预分配缓冲区
    if ret.data_ptr() != output.data_ptr():
        output[:real_num_tokens].view(ret.shape).copy_(ret)
    # ... HIP 特殊处理
    # 将 LSE 填充到 bucket 形状，保证 CUDA 图重放时形状稳定
    if lse is not None and lse.shape[0] != output.shape[0]:
        padded_lse = lse.new_zeros((output.shape[0], *lse.shape[1:]))
        padded_lse[:real_num_tokens].copy_(lse)
        lse = padded_lse
    return lse
```

test/registered/unit/layers/test_radix_attention.py

新增针对 extra-kwargs 路径的 LSE 回归测试。

```
# test/registered/unit/layers/test_radix_attention.py
def test_extra_kwargs_path_returns_bucket_shaped_lse(self):
    attention_layer = SimpleNamespace()
    context = self._new_impl_context([attention_layer])
    backend = _RecordingAttentionBackend()
    query = torch.zeros((4, 2, 3))
    with (
        patch.object(radix_attention_module, "get_tc_piecewise_forward_context", return_value=
        context),
```

```
        patch.object(radix_attention_module, "get_attn_backend", return_value=backend),
    ):
        lse = radix_attention_module.attention_with_output_extra_kwargs(
            query, query, query, torch.empty_like(query), False, 0, {"return_lse": True}
        )
    # 验证真实 2 个 token 的 LSE 为 7，填充的 2 个 token 为 0
    self.assertEqual(lse.tolist(), [[7, 7], [7, 7], [0, 0], [0, 0]])
```

评论区精华

Review 仅有一条批准评论，无实质讨论。评论中的 CI 重跑指令表明有测试失败，但最终被判定为 runner 问题。

- CI 重跑 (other): 最终 CI 通过，被认为 runner 问题。

风险与影响

- 风险：主要风险在于对核心注意力路径的修改，但变更仅在请求 return_lse 时激活，对 tensor-only 路径无影响。CUDA 图场景下 LSE 的填充逻辑需要确保形状正确，测试已覆盖。潜在风险包括：return_lse 键可能被误用，或某些后端未正确返回 LSE 元组，但通过 assert 进行了保护。
- 影响：影响范围限定在需要同时使用 extra-kwargs 和 LSE 的注意力调用，主要涉及相对偏置等特殊后端的场景。对普通用户无影响，团队内其他开发者可能受益于该修复。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #36909 [mem_cache] Carry swa_evicted_seqlen into SWARadixCache.cache_unfinished_req: 同为 RadixCache/ 注意力相关修复，涉及缓存和注意力路径。
- PR #36704 Refactor JIT kernel and expert-pack directory layout: 重构涉及注意力内核的 JIT 布局，与 RadixAttention 间接相关。