PR #31838 完整报告

sgl-project/sglang

Fix pad-row top-k masking with custom_routing_function under DP attention

合并时间：2026-07-22 02:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31838

执行摘要

- 一句话：修复 custom routing 下 pad-row mask 缺失及 prefill replay 中 num_token_non_padded 计算错误
- 推荐动作：值得精读：展示了在 CUDA-graph 动态形状与 MoE 路由交互时的边界处理，尤其是 post_fill 钩子的运用方式以及如何利用 host int 避免 replay 时 host-to-device copy。设计决策清晰，代码注释详细，是学习 SGLang MoE + 图执行系统的良好案例。

功能与动机

select_experts 在 custom_routing_function 分支断言 num_token_non_padded is None，任何使用自定义路由函数的模型都无法传入该参数。在带有 CUDA-graph padding 的 DP attention 下，padded row 的 router logits 是垃圾数据，若不 mask，padded row 会保留 unmasked top-k expert ids，导致 per-expert dispatch 计数倾斜并可能溢出 fused EP MoE 内核的 per-expert buffer，最终输出 NaN 或错误结果。

实现拆解

1. 删除自定义路由分支的断言：在 python/sglang/srt/layers/moe/topk.py 的 select_experts 中删除 assert num_token_non_padded is None，并添加注释说明 padding-unaware 的自定义路由输出在后处理 _post_process_topk_ids 中会被 mask（CUDA 上 padded row 设为 -1，HIP 上设为 0 并 zero 权重）。
2. 提取 attn-TP shard 边界计算：在 python/sglang/srt/model_executor/forward_batch_info.py 中新增 _attn_tp_local_shard_bounds 函数，返回当前 attn-TP rank 的 tokens_per_rank 和 rank_offset；重构 compute_local_num_token_non_padded 使用新函数；新增整数版本 compute_local_num_token_non_padded_cpu 用于 replay 时避免 host-to-device copy。
3. 在 prefill registry 添加 post_fill 钩子：在 python/sglang/srt/model_executor/cuda_graph_buffer_registry.py 的 build_prefill_registry 中，当 enable_num_token_non_padded 时注册 _prefill_num_token_non_padded_post_fill 作为 num_token_non_padded 槽位的 post_fill。该钩子利用 fb.num_token_non_padded_cpu（全局未调整计数）和 ctx.padded_num_tokens（bucket 大小）重新计算 local count，仅在 require_gathered_buffer=True 且 enable_prefill_cp=False 时生效。同时为 build_prefill_registry 新增 require_gathered_buffer 和 enable_prefill_cp 参数。

4. 在 prefill CUDA-graph runner 传入新参数：在 python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py 中，向 build_prefill_registry 传递 require_gathered_buffer 和 enable_prefill_cp。
5. 新增测试覆盖：在 test/registered/unit/model_executor/test_cuda_graph_buffer_registry.py 中增加 TestPrefillNumTokenNonPaddedPostFill 测试类，验证不同 attn-tp rank 下 post_fill 正确使用 bucket shard 而非原始 FB 值（rank 0 应返回 bucket/attn_tp，rank 1 应正确 clamp 到真实 padded 行数）；在 test/registered/moe/test_topk_padded_region.py 中增加 TestSelectExpertsCustomRoutingPadMask 测试类，验证 select_experts 接受 num_token_non_padded 且 custom router 输出中的 padded row 被 mask 为 -1，真实行保持不变。

关键文件：

- python/sglang/srt/layers/moe/topk.py（模块 MoE 路由；类别 source；类型 core-logic；符号 select_experts）：核心 bug 修复：删除 custom_routing_function 分支的断言，允许传递 num_token_non_padded，并依赖后处理 mask padded region。
- python/sglang/srt/model_executor/forward_batch_info.py（模块 批信息；类别 source；类型 data-contract；符号 _attn_tp_local_shard_bounds, compute_local_num_token_non_padded_cpu）：新增辅助函数 _attn_tp_local_shard_bounds 和整数版本 compute_local_num_token_non_padded_cpu，重构原函数以复用公共逻辑。
- python/sglang/srt/model_executor/cuda_graph_buffer_registry.py（模块 图注册表；类别 source；类型 data-contract；符号 _prefill_num_token_non_padded_post_fill）：新增 prefill registry 的 post_fill 钩子 _prefill_num_token_non_padded_post_fill，在 replay 时根据 bucket 大小重新计算 local num_token_non_padded。
- test/registered/unit/model_executor/test_cuda_graph_buffer_registry.py（模块 图注册表测试；类别 test；类型 test-coverage；符号 TestPrefillNumTokenNonPaddedPostFill, _fill, test_rank0_uses_bucket_shard_not_raw_localized_value, test_rank1_masks_exactly_the_true_pads）：新增 TestPrefillNumTokenNonPaddedPostFill 测试类，验证 prefill replay 时 num_token_non_padded 的 post_fill 正确性。
- test/registered/moe/test_topk_padded_region.py（模块 MoE Padded 区域测试；类别 test；类型 test-coverage；符号 TestSelectExpertsCustomRoutingPadMask, test_padded_tail_masked_after_custom_routing, _degenerate_router）：新增 TestSelectExpertsCustomRoutingPadMask 测试类，验证 select_experts 在 custom_routing_function 下正确 mask padded row 为 -1。
- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（模块 预填充执行器；类别 source；类型 configuration）：在构建 prefill registry 时传入 require_gathered_buffer 和 enable_prefill_cp 参数，激活 post_fill 钩子。

关键符号：select_experts, compute_local_num_token_non_padded_cpu, _attn_tp_local_shard_bounds, _prefill_num_token_non_padded_post_fill, build_prefill_registry

关键源码片段

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

核心 bug 修复：删除 custom_routing_function 分支的断言，允许传递 num_token_non_padded，并依赖后处理 mask padded region。

```
# python/sglang/srt/layers/moe/topk.py
# 在 select_experts 函数中，原本自定义路由分支有断言：
# assert num_token_non_padded is None, ...
# 删除该断言并添加注释：
else:
    # custom_routing_function 本身对 padding 不感知，其 padded row 输出是垃圾数据。
    # 但这是安全的，因为下面的 _post_process_topk_ids 会在 logical->physical 重映射后
    # 将 num_token_non_padded 及之后的行 mask 掉（CUDA 上 topk_ids 设为 -1，
    # HIP 上设为 0 并 zero 权重）。
    assert not apply_routed_scaling_factor_on_output, "Not implemented"
    topk_weights, topk_ids = custom_routing_function(
        hidden_states=hidden_states,
        gating_output=router_logits,
        topk=topk_config.top_k,
        renormalize=topk_config.renormalize,
    )
    # 后续 shared path 会调用 _post_process_topk_ids 进行 pad mask
```

python/sglang/srt/model_executor/forward_batch_info.py

新增辅助函数 _attn_tp_local_shard_bounds 和整数版本 compute_local_num_token_non_padded_cpu，重构原函数以复用公共逻辑。

```
# python/sglang/srt/model_executor/forward_batch_info.py

def _attn_tp_local_shard_bounds(num_tokens_per_dp: int) -> Tuple[int, int]:
    """返回当前 attn-TP rank 的连续 shard 的 (tokens_per_rank, rank_offset)。"""
    parallel = get_parallel()
    tokens_per_rank = num_tokens_per_dp // parallel.attn_tp_size
    return tokens_per_rank, tokens_per_rank * parallel.attn_tp_rank


def compute_local_num_token_non_padded(
    global_num_token_non_padded: torch.Tensor,
    num_tokens_per_dp: int,
) -> torch.Tensor:
    """将全局计数（当前 DP rank 内）转为本地 attn-TP rank 的计数。"""
    tokens_per_rank, rank_offset = _attn_tp_local_shard_bounds(num_tokens_per_dp)
    return torch.clamp(
        global_num_token_non_padded - rank_offset,
        0,
        tokens_per_rank,
    )
```

```
def compute_local_num_token_non_padded_cpu(
    global_num_token_non_padded: int,
    num_tokens_per_dp: int,
) -> int:
    """整数版本，用于 replay 时直接在 host 计算，然后通过 Tensor.fill_ 写入 GPU buffer。"""
    tokens_per_rank, rank_offset = _attn_tp_local_shard_bounds(num_tokens_per_dp)
    return min(max(global_num_token_non_padded - rank_offset, 0), tokens_per_rank)
```

python/sglang/srt/model_executor/cuda_graph_buffer_registry.py

新增 prefill registry 的 post_fill 钩子 _prefill_num_token_non_padded_post_fill，在 replay 时根据 bucket 大小重新计算 local num_token_non_padded。

```
# python/sglang/srt/model_executor/cuda_graph_buffer_registry.py
# 在 build_prefill_registry 函数内的 slot 注册部分
if enable_num_token_non_padded:
    from sglang.srt.model_executor.forward_batch_info import (
        compute_local_num_token_non_padded_cpu,
    )

    def _prefill_num_token_non_padded_post_fill(buf, fb, ctx):
        # FB tensor 中的 num_token_non_padded 是基于 RAW 长度本地化的，
        # 但 replay 会将 token 数填充到 capture bucket，从而移动了 attn-TP shard 边界。
        # 如果直接复制 FB 的值，当 raw < bucket 时，pad mask 会错误地覆盖真实 token。
        # 因此需要根据 bucket 大小（ctx.padded_num_tokens）重新计算本地计数。
        # 该逻辑仅在使用 gathered buffer 且未启用 prefill context parallelism 时生效。
        if require_gathered_buffer and not enable_prefill_cp:
            buf.fill_(
                compute_local_num_token_non_padded_cpu(
                    global_num_token_non_padded=fb.num_token_non_padded_cpu,
                    num_tokens_per_dp=ctx.padded_num_tokens,
                )
            )

    slots.append(
        GraphSlot(
            "num_token_non_padded",
            lambda _bs2, _mt: (1,),
            torch.int32,
            axis="none",
            post_fill=_prefill_num_token_non_padded_post_fill,
        )
    )
```

评论区精华

PR 无公开 review 讨论，作者自行测试后合并。作者在最后一条评论中提到内部测试了 kl + stress accuracy，CI 全通过。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 核心路径变更：select_experts 是 MoE 路由关键函数，删除断言并依赖后处理 mask 可能让旧的调用方在未传入 num_token_non_padded 时行为不变，但任何忘记传入 num_token_non_padded 的 custom router 场景将失去保护（之前断言会直接报错，现在 silent 地产生垃圾输出）。不过后处理 _post_process_topk_ids 实际上在 shared path 已处理这种情况。
  - CUDA-graph replay 依赖：_prefill_num_token_non_padded_post_fill 仅在 require_gathered_buffer 且 enable_prefill_cp=False 时触发，若未来其他图模式（如 breakable prefill context parallelism）未正确设置这两个参数，可能导致仍使用错误的 local count。
  - HIP 平台测试跳过：TestSelectExpertsCustomRoutingPadMask 跳过了 HIP 平台，DP attention + HIP 场景可能未被覆盖，不过 HIP 路径有独立 mask 逻辑。
  - 性能影响：post_fill 仅执行整数计算和 fill_，影响极小。
  - 影响：影响范围：主要影响使用自定义路由函数（custom_routing_function）且启用 DP attention + CUDA-graph padding 的模型（如内部部署的模型）。这些用户在升级后应不再遇到 NaN 输出。对于未使用 custom_routing_function 或 DP attention 的用户，无行为变化。影响程度：修复了正确的功能性 bug，提升稳定性。测试覆盖了回归场景。
  - 风险标记：自定义路由路径变更，CUDA-graph replay 依赖，缺少 HIP 测试覆盖

关联脉络

- PR #31682 Turn on breakable prefill cuda graph for dp attention by default: 该 PR 默认启用了 breakable prefill CUDA graph，使得本 PR 修复的 prefill replay num_token_non_padded 重新计算问题暴露并需要修复。
- PR #31835 Negotiate PrefillDelayer only after KV-budget admission checks: 同为调度相关的 bugfix，但无直接技术关联，仅同属调度 /MoE 主干。