

PR #39574 完整报告

sgl-project/sglang

[misc] Fix tool-call index, graph padded-row count, and prefill-graph input_embeds refresh

合并时间: 2026-09-16 07:37

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39574>

执行摘要

- 一句话: 修复工具调用序号、图填充计数契约、prefill 嵌入刷新等五项缺陷
- 推荐动作: 值得精读。两个设计决策有学习价值: ① 用 "槽位存在性" 表达数据契约——槽位不存在即契约无效, 让 CUDA graph 静态批与 eager 批在两条路径上共享同一不变量, 避免再次出现 "graph 携带 stale 状态、eager 携带 None" 的分叉; ② replay 闭包处理输入槽位的模式——能从 kwargs、位置参数逐步回退到现场计算 (get_input_embeddings), 可复用于其他 graph runner 的输入刷新。建议合并前向作者确认非 EP 下其他 num_token_non_padded 读取点的审计情况。

功能与动机

PR body 逐条给出了五个独立缺陷的机理:

- 1) 非流式 `_process_tool_calls` 按 detector 的 `tool_index` 编号, 而该值对多数 detector 是工具在请求中的位置, 同一工具两次调用都得到 `index=0`, 与流式 `delta` 的 0/1 编号不一致;
- 2) decode graph runner 无条件分配 `num_token_non_padded` 槽位、capture 时写入、仅 EP 下 replay 刷新, 导致非 EP 时静态 `ForwardBatch` 携带 capture 时的陈旧计数, 而 eager batch 的 `ForwardBatch.init_new` 只在 EP 下填充该值;
- 3) 标准 MoE 路径未像 a2a 路径那样把 padded rows 路由到 expert -1;
- 4) 外层 forward 不传 `input_embeds` 时 prefill 图回放闭包不刷新槽位, 图会读取上一批次嵌入;
- 5) FlashInfer $\geq 0.6.18$ 按内容定型 `do_finalize=False` 的权重缓冲, 旧的 `fp32`→`bf16` 位重解释只对旧版本正确, 会损坏真实 `fp32` 权重。

实现拆解

按 5 个独立修复拆分实现:

1. 工具调用序号修复 (`python/sglang/srt/entrypoints/openai/serving_chat.py`):
`_process_tool_calls` 将 `for call_info in call_info_list` 改为 `for index, call_info in enumerate(call_info_list)`, `ToolCall.index` 直接使用调用序号, 替代 `getattr(call_info, "tool_index", None)`, 保留原有 `_process_tool_call_id` 与 `finish_reason` 逻辑。配套回归测试 `test_serving_chat.py::test_non_streaming_tool_call_index_is_the_call_ordinal` 用 mock parser 构造同一工具两次调用 (`tool_index` 均为 0), 断言输出 `index` 为 `[0, 1]` 且 `arguments` 顺序保留。

2. num_token_non_padded 契约收紧 (runner_utils/buffers.py、runner/base_runner.py、runner/decode_cuda_graph_runner.py) : DecodeInputBuffers 字段类型由 torch.Tensor 改为 Optional[torch.Tensor]; DecodeInputBuffers.create 与 _allocate_decode_buffers 仅在 enable_num_token_non_padded() 为真时分配槽位; _dummy_run 与 decode 图回放的写入全部改为 if buffers.num_token_non_padded is not None 判空。外层判空后, decode runner 内 SP 本地计数换算省略了原先重复的 enable_num_token_non_padded() 判断 (逻辑等价)。
3. MoE 标准路径屏蔽填充行 (models/deepseek_v2.py) : DeepseekV2MoE.forward 从 forward_batch 提取 num_token_non_padded 并透传给 forward_normal 与 forward_normal_dual_stream, 最终传入 self.topk(..., num_token_non_padded=...), 与 forward_deepep 既有行为对齐; 同时双流变体在 is_in_breakable_cuda_graph() 为真时跳过, 与 hc 路径的既有守卫保持一致。
4. prefill 图回放刷新嵌入 (runner/prefill_cuda_graph_runner.py) : _execute_body_capture 内嵌的 replay_layer_forward 在既无 input_embeds kwarg 也无第 4 个位置参数时, 用 args[0] (或 layer_kwargs["input_ids"]) 调用 model.get_input_embeddings() 现场计算嵌入并拷入静态槽位。
5. deferred finalize ABI 适配 (layers/moe/moe_runner/flashinfer_trtllm.py) : _make_deferred_finalize_output 删除 fp32->bf16 位重解释, 改为校验 dtype 属于 {bf16, fp32} 后直接透传。附带清理: 删除 decode runner 中未被使用的 ngram_embedding_n/k 属性。

验证配套: 除新增回归测试外, 跑通 test_serving_chat.py、model_executor 单测、batch_overlap、dp-attention、LoRA MoE、EPLB 单测; Kimi-VL-A3B test_vision_openai_server_a.py (混合 image/text batch、decode CUDA graph、EP off) 早期修订版触发 fused MoE kernel 非法内存访问, 当前 head 通过且输出与 main 一致, 混合批 client 重复回放 6 次。

关键文件:

- python/sglang/srt/model_executor/runner/decode_cuda_graph_runner.py (模块 图执行器; 类别 source; 类型 data-contract; 符号 DecodeCudaGraphRunner) : decode CUDA graph runner 是 num_token_non_padded 契约修复的核心: 刷新逻辑改为槽位判空, 非 EP 下不再写入 stale 计数, 直接消除 Kimi-VL-A3B 场景的 fused MoE kernel 非法内存访问; 同时清理死代码 ngram_embedding_n/k。
- python/sglang/srt/model_executor/runner_utils/buffers.py (模块 静态缓冲; 类别 source; 类型 data-contract; 符号 DecodeInputBuffers.create) : DecodeInputBuffers 的数据契约变更入口: 字段类型改为 Optional[torch.Tensor], create 仅在 EP 下分配槽位, 从源头保证 graph 批与 eager 批遵循同一规则。
- python/sglang/srt/model_executor/runner/base_runner.py (模块 执行器基类; 类别 source; 类型 data-contract; 符号 _allocate_decode_buffers, _dummy_run) : eager/dummy 侧的配套契约修复: _allocate_decode_buffers 条件化分配、_dummy_run 写入判空, 保证 dummy run 与真实回放路径一致。
- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py (模块 图执行器; 类别 source; 类型 data-contract; 符号 _execute_body_capture,

replay_layer_forward) : prefill 图回放闭包在 ie 缺失时由 input_ids 现场计算嵌入, 修复图读取上一批次嵌入的陈旧数据问题。

- python/sglang/srt/models/deepseek_v2.py (模块 MoE 路由; 类别 source; 类型 data-contract; 符号 DeepseekV2MoE.forward, forward_normal, forward_normal_dual_stream) : DeepSeekMoE 标准路径 (forward_normal / forward_normal_dual_stream) 透传 num_token_non_padded 到 top-k, 使填充行在 EP + 标准 dispatcher 下路由到 expert -1, 与 a2a 路径一致。
- python/sglang/srt/layers/moe/moe_runner/flashinfer_trtllm.py (模块 MoE 内核; 类别 source; 类型 core-logic; 符号 _make_deferred_finalize_output) : deferred finalize 的 ABI 适配: 删除会损坏真实 fp32 权重的旧位重解释, 改为按内容校验 bf16/fp32, 适配 FlashInfer >= 0.6.18。
- python/sglang/srt/entrypoints/openai/serving_chat.py (模块 聊天服务; 类别 source; 类型 core-logic; 符号 _process_tool_calls) : 非流式工具调用序号修复的源码入口: ToolCall.index 从 detector 的 tool_index (工具在请求中的位置) 改为调用序数, 与流式 delta 对齐。
- test/registered/unit/entrypoints/openai/test_serving_chat.py (模块 服务测试; 类别 test; 类型 test-coverage; 符号 test_non_streaming_tool_call_index_is_the_call_ordinal) : 新增回归测试 test_non_streaming_tool_call_index_is_the_call_ordinal, 用 mock parser 验证同一工具两次调用的 index 为 [0, 1], 防止序号退化。

关键符号: _process_tool_calls, DecodeInputBuffers.create, _allocate_decode_buffers, _dummy_run, replay_layer_forward, DeepseekV2MoE.forward, DeepseekV2MoE.forward_normal, DeepseekV2MoE.forward_normal_dual_stream, _make_deferred_finalize_output

关键源码片段

python/sglang/srt/model_executor/runner/decode_cuda_graph_runner.py

decode CUDA graph runner 是 num_token_non_padded 契约修复的核心: 刷新逻辑改为槽位判空, 非 EP 下不再写入 stale 计数, 直接消除 Kimi-VL-A3B 场景的 fused MoE kernel 非法内存访问; 同时清理死代码 ngram_embedding_n/k。

```
# decode 图回放前的缓冲准备阶段: 按槽位是否存在决定是否刷新计数。
# 该槽位只在 enable_num_token_non_padded() (即 EP 场景) 下分配,
# 非 EP 时保持 None, 避免静态 ForwardBatch 携带 capture 时的陈旧
# 计数, 进而导致 fused MoE kernel 读取越界 (Kimi-VL-A3B 实测崩溃)。
attn_tp_sharded = self.model_runner.attn_tp_sequence_sharded(num_tokens)
if buffers.num_token_non_padded is not None:
    # 槽位存在 = 契约有效: 写入全局 token 数, 并视 attn-TP 分片
    # 情况换算为本地计数 (SP on 时每个 rank 只持有分片后的 token)。
    buffers.num_token_non_padded[...] = num_tokens
    if not self.enable_prefill_cp and attn_tp_sharded:
        local = compute_local_num_token_non_padded(
            global_num_token_non_padded=buffers.num_token_non_padded,
            num_tokens_per_dp=num_tokens,
```

```

        sharded=True,
    )
    buffers.num_token_non_padded.copy_(local)

```

python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py

prefill 图回放闭包在 ie 缺失时由 input_ids 现场计算嵌入，修复图读取上一批次嵌入的陈旧数据问题。

```

def replay_layer_forward(*args, **layer_kwargs):
    # 被捕获的 prefill body 图从静态 input_embeds 槽位读取激活；外层
    # model.forward (eager 执行) 会把实时嵌入作为第 4 个位置参数或
    # input_embeds kwarg 传给 layer_model.forward。回放前必须把当前
    # 请求的嵌入拷入槽位，否则图会读到上一批次的嵌入。
    if self.buffer_registry.has_slot("input_embeds"):
        ie = layer_kwargs.get("input_embeds")
        if ie is None and ie_idx is not None and len(args) > ie_idx:
            ie = args[ie_idx]
        if ie is None:
            # 纯文本 batch 外层不传嵌入时，直接由 input_ids 现场计算，
            # 与多模态 batch 的 eager 路径语义保持一致。
            input_ids = args[0] if args else layer_kwargs["input_ids"]
            ie = self.model_runner.model.get_input_embeddings()(input_ids)
        self.buffer_registry.get_slot("input_embeds").slice_for(
            1, static_num_tokens
        )[: ie.shape[0]].copy_(ie)
    hs = self.backend.replay(shape_key, static_forward_batch, **kwargs)
    return _slice_output_rows(hs, raw_num_tokens) if full_path else hs

```

python/sglang/srt/entrypoints/openai/serving_chat.py

非流式工具调用序号修复的源码入口：ToolCall.index 从 detector 的 tool_index（工具在请求中的位置）改为调用序数，与流式 delta 对齐。

```

tool_calls = []
# 按调用序数编号，与流式 delta 保持一致；
# tool_index 表示工具在请求中的位置，同一工具被调用两次时
# 两个调用都是 0，不能再作为 index 使用。
for index, call_info in enumerate(call_info_list):
    tool_id = self._process_tool_call_id(call_info, history_tool_calls_cnt)
    tool_calls.append(
        ToolCall(
            id=tool_id,
            index=index,
            function=FunctionResponse(
                name=call_info.name,
                arguments=call_info.parameters,
            ),
        )
    )
# 后续 finish_reason 处理逻辑保持不变：

```

有调用时置为 tool_calls, 无调用时维持原类型。

评论区精华

本 PR 无人工 review 评论 (review_comments_count = 0) , 4 条 Issue 评论均为 CI 指令与机器人结果。最有信息量的讨论沉淀在 PR body 与 CI 首跑失败中:

- 作者在 PR body 第 2 项给出契约统一表述: "Graph and eager batches now follow the same rule: no count unless it is live."——即槽位存在性即契约有效性, 静态批与 eager 批共享同一不变量。
- 第一次 /rerun-test 中 1-gpu-h100 的 test_vision_openai_server_a.py 失败 (机器人首次报告 [🔗](#)) , 重跑后通过; 作者定位到早期修订版因图 ForwardBatch 携带 stale num_token_non_padded 计数, 在 fused MoE kernel 中触发 illegal memory access, 当前 head 输出与 main 完全一致。
- deferred finalize 的版本权衡: 旧位重解释只对 " 以 routing_logits dtype 分配缓冲却写入 bf16" 的 FlashInfer 版本正确; FlashInfer >= 0.6.18 (flashinfer #3595) 按内容定型, 必须改为按内容校验 bf16/fp32。
- CI 首跑失败与 stale count 引发的非法内存访问 (testing): 当前 head 通过且输出与 main 完全一致, 混合批 client 重复回放 6 次; 修复正是本 PR 的第 2/3 项。
- num_token_non_padded 的 graph/eager 契约统一 (correctness): 槽位仅在 enable_num_token_non_padded() 时分配, 所有写入点判空; graph 与 eager 遵循同一规则: 计数无效时即 None (PR body 原文 no count unless it is live) 。
- deferred finalize 权重 dtype 的版本适配 (correctness): 改为按内容校验 bf16/fp32 并直接透传, moe_finalize_fuse_shared 本就接受两种 dtype。

风险与影响

- 风险:
 1. 数据契约变更 (主要风险) : num_token_non_padded 变为 Optional 后, 非 EP 路径上所有直接解引用该槽位的代码会得到 None。本 PR 覆盖了 decode runner、dummy run、MoE top-k 传递, 但仓库中其他读取点 (调度器、DP attention、PP 统计等) 需审计; 建议合并后跑 EP/ 非 EP 双档回归矩阵。
 2. 路由行为变更: EP 开 + 标准 dispatcher 时, CUDA graph 填充行由 " 路由到真实专家 " 变为 " 路由到 expert -1", expert 分布统计与负载均衡特征会变化。这是有意修复, 但可能影响在线集群的 EPLB 表现, 需观察。
 3. FlashInfer 版本依赖: 若实际部署 FlashInfer < 0.6.18, 仍可能返回 " 按 routing_logits dtype 分配、写入 bf16 的 fp32 缓冲 ", 新逻辑不再做兼容位重解释而是直接放行 fp32。PR 说明 moe_finalize_fuse_shared 双 dtype 支持, 但仍依赖该 kernel 对 fp32 的最终处理正确, 建议验证目标版本组合。
 4. 额外嵌入计算: prefill 回放仅在 ie 缺失时调用 get_input_embeddings(), 为纯文本 batch 新增一次嵌入计算; 该调用发生在 eager 回放闭包内、不进入图捕获, GPU 开销可控但属于新路径。

5. 工具调用行为变更：非流式多调用同一工具的 index 语义从 " 均为 0 " 变为 " 调用序数 "，依赖旧语义的客户端会受影响（这是与流式对齐的正确性修复）。- 影响：对用户：DeepSeek 系列（含 DSV4）在 EP + CUDA graph 下的推理稳定性与正确性提升，非流式工具调用序号与流式响应一致。对系统：消除一类静态批陈旧计数导致的 GPU 非法内存访问（属于稳定性修复）；非 EP 路径少分配一个 int32 标量，内存影响可忽略。对团队：单文件小改动、独立 commit 便于回滚与逐项 review，但 num_token_non_padded 契约变化需要 EP/ 非 EP 双档回归确认无其他解引用点遗漏。- 风险标记：CUDA graph 静态批契约变更，EP/ 非 EP 行为差异需回归，依赖 FlashInfer 版本行为，填充行路由行为变更，非流式工具调用 index 行为变更

关联脉络

- PR #38176 keeping router GEMM in fp32 for deterministic inference (DeepSeek V3/V4): 同改 python/sglang/srt/models/deepseek_v2.py, 同属 DeepSeek V3/V4 推理正确性修复线：一个管 router GEMM 精度、一个管填充行路由，两者共同保障 EP + CUDA graph 下 MoE 确定性。
- PR #39328 Fix first-token metadata and reused attention-layer indexing: 同为 runner 组件正确性修复 (cuda_graph_setup.py)，处理 " 静态图状态复用前一次即时数据 " 这一共同模式，与本 PR 的 input_embeds 槽位陈旧问题同源。
- PR #39474 [qwen 3.8 next] reuse old cuda stream instead of endlessly creating streams: 同为 CUDA graph 捕获语义修复（流复用），与本 PR 的 capture/replay 正确性主题呼应，均属于 runner 层图执行稳定性建设。
- PR #38984 [Intra-node PD][DSV4] Pack all layers into one batch for INTRA_NODE_NVLINK path: DSV4 家族性能 / 正确性演进之一，与本 PR 同属 DeepSeek V4 在 PD/EP 场景下的稳定性建设，可对照阅读 MoE 路径的演进脉络。