

PR #35042 完整报告

sgl-project/sglang

Fix sconv track refresh on graph capture

合并时间: 2026-08-17 15:51

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35042>

执行摘要

- 一句话: 修复 CUDA graph 捕获时 sconv track 索引刷新崩溃
- 推荐动作: 值得精读。虽然只有 10 行有效改动, 但它演示了如何修复“由另一修复引入的可达性回归”而不破坏原有不变量。重点学习 `on_graph_path` 双重守卫的推导过程, 以及作者用精度对照表排除三个候选方案的决策方法。后续为 MTP + Unified Radix Cache 补充 bit-exact 测试时, 应把本场景固化为回归用例。

功能与动机

Issue #35039 报告: Serving Inkling with MTP 在 draft-extend CUDA graph 捕获阶段崩溃, `forward_batch.extend_prefix_lens` 为 `None`。PR body 进一步说明根因: 该读取自 #33023 引入但一直不可达, #34043 使捕获路径填充 inert track buffer 并继续执行, 导致捕获仍启动 track scatter, 使该读取首次可达; 而 `capture batch` 直接构造、不走 `ForwardBatch.init_new`, 所以前缀长度缺失。

实现拆解

1. 定位崩溃点: `python/sglang/srt/layers/attention/linear/inkling_sconv_backend.py` 的 `_refresh_track_conv_indices` 直接读取 `forward_batch.extend_prefix_lens.shape[0]`, 在 draft-extend 捕获路径中该字段为 `None`, 抛出 `AttributeError`。
2. 引入安全兜底: 新增局部变量 `prefix_lens = forward_batch.extend_prefix_lens`, 当 `prefix_lens is None and on_graph_path` 时, 用 `torch.zeros_like(forward_batch.mamba_track_seq_lens)` 替代。选择零值而非提前返回的原因: 捕获 batch 的各行被 inert mask 覆盖, 零前缀长度使 `lens_to_track` 取整后仍落在合法索引区间。
3. 统一后续计算: 将 live 的 min 计算、`lens_to_track` 的减法统一改用局部 `prefix_lens`, 保证图路径与非图路径共用同一套窗口索引逻辑。
4. 保持不变量: 非图路径 (`on_graph_path=False`) 下 `None` 仍会触发原生异常, 保留“缺失前缀长度意味着真实异常”的语义, 避免用编造的零前缀掩盖 live 路径的问题。
5. 测试与 CI 配套: 本 PR 未新增测试文件, 依赖现有 bit-exact KL 一致性测试 (`test_unified_radix_cache_kl_hybrid_bitexact.py`、`test_inkling.py`、`test_inkling_small_nvfp4.py`) 做回归验证, 作者通过 `/rerun-test` 在 1-gpu-h100 与 4-gpu-b200 上确认通过。

关键文件：

- python/sglang/srt/layers/attention/linear/inkling_sconv_backend.py（模块 注意力层；类别 source；类型 core-logic；符号 _refresh_track_conv_indices）：核心修复文件：_refresh_track_conv_indices 在 CUDA graph 捕获路径上对 extend_prefix_lens 做安全兜底，修复 Inkling + MTP 启动崩溃。

关键符号：_refresh_track_conv_indices

关键源码片段

python/sglang/srt/layers/attention/linear/inkling_sconv_backend.py

核心修复文件：_refresh_track_conv_indices 在 CUDA graph 捕获路径上对 extend_prefix_lens 做安全兜底，修复 Inkling + MTP 启动崩溃。

```
def _refresh_track_conv_indices(
    self, forward_batch: ForwardBatch, on_graph_path: bool
):
    """刷新用于前缀缓存的卷积窗口快照位置：取最后 ``conv_kernel - 1`` 个 token、
    最靠近 ``mamba_cache_chunk_size`` 边界的一个完整窗口。

    尾部填充必须为 0 而非遗留旧值：被捕获的 gather 会读取全部 ``batch_size`` 行，
    而 track 长度只覆盖活跃请求，任何可能被读到的行都必须索引到本次 replay 的
    token buffer 之内。
    """
    if forward_batch.mamba_track_mask is None:
        if not on_graph_path:
            # 非图路径：没有追踪元数据就直接清空，提前返回
            self.sconv_metadata.track_conv_indices = None
            return
        # 图捕获路径：仍要启动 track scatter（见 __init__ 中的 inert buffer），
        # 用预分配的 inert 张量占位，保证捕获形状稳定
        rows = forward_batch.batch_size
        forward_batch.mamba_track_mask = self._graph_track_inert_mask[:rows]
        forward_batch.mamba_track_indices = self._graph_track_inert_indices[:rows]
        forward_batch.mamba_track_seqlens = self._graph_track_inert_seqlens[:rows]

    rows = forward_batch.batch_size
    query_start_loc = self.sconv_metadata.query_start_loc

    # 捕获期 batch 是直接构造的（不走 ForwardBatch.init_new），因此一定没有
    # extend_prefix_lens；其行会被 mask 掉，全零可保证索引仍在界内。
    # 而重放或 eager 路径的 batch 必然携带前缀长度，若缺失应直接报错，
    # 不能拿编造的零前缀去 track。
    prefix_lens = forward_batch.extend_prefix_lens
    if prefix_lens is None and on_graph_path:
        prefix_lens = torch.zeros_like(forward_batch.mamba_track_seqlens)

    live = min(
```

```

        rows,
        forward_batch.mamba_track_seqlens.shape[0],
        prefix_lens.shape[0],
    )

    # 需要快照的窗口长度 = 现有 seqlen 减去扩展前缀长度，再对齐到 cache chunk
    lens_to_track = forward_batch.mamba_track_seqlens[:live] - prefix_lens[:live]
    chunk_aligned = (
        lens_to_track // self.mamba_cache_chunk_size
    ) * self.mamba_cache_chunk_size
    start_indices = query_start_loc[:live] + chunk_aligned - self.conv_state_len

    if on_graph_path:
        # 图路径必须写入固定形状的预分配 buffer，否则捕获失败
        assert rows <= self._graph_track_conv_indices.shape[0], (
            f"track-index buffer too small for a captured shape: rows={rows} "
            f"vs bound {self._graph_track_conv_indices.shape[0]}"
        )
        out = self._graph_track_conv_indices[:rows]
    else:
        out = torch.empty(
            (rows, self.conv_state_len),
            dtype=torch.int64,
            device=start_indices.device,
        )

    # 广播窗口偏移生成索引矩阵，并 clamp 到合法区间
    torch.add(
        start_indices.unsqueeze(-1).to(torch.int64),
        self._track_window_offsets,
        out=out[:live],
    )

    # 1 元素张量，绝不为 -1：0-d 转 Python 会引发同步
    torch.clamp(
        out[:live],
        min=self._track_index_floor,
        max=query_start_loc[-1:].to(torch.int64) - 1,
        out=out[:live],
    )

    # 填充尾部为 0，避免 gather 读到越界索引
    if live < rows:
        out[live:].zero_()

```

评论区精华

PR 无外部 review 评论，作者在 body 中给出了 5 组精度对照实验，核心权衡如下：

- 回滚 #34043 的 sconv hunk 虽能消除崩溃，但 #34043 修复了真实状态损坏，因此不是候选修复；

- 仅用 `extend_prefix_lens` is None 守卫会放宽不变量：任何未来 live 路径缺前缀时会静默以零 track；
- 以 `inert-fill` 分支为守卫不可靠：inert fill 会把 `mamba_track_mask` 写回 batch，而 capture 会在 warmup 和 capture 间复用同一个 ForwardBatch，第二次调用会命中另一分支；
- 最终采用 `None` 与 `on_graph_path` 双重守卫，兼顾捕获路径可用性与非图路径失败语义。
- capture 路径缺失 `prefix_lens` 时如何安全降级 (design): 采用 `prefix_lens` is None and `on_graph_path` 双重守卫，图路径用零填充保证索引在界内，非图路径仍保持失败语义。

风险与影响

- 风险：
 1. 捕获路径语义依赖 inert mask：兜底零值只在捕获 batch 的行被 mask 掉时安全；若未来新增捕获路径未做 mask，可能生成错误的 track 索引（当前所有 capture 场景均满足前提）。
 2. 不变量保持：非图路径 `None` 仍会抛错，避免静默错误，但这也意味着任何未来走非图路径且缺失前缀的调用依然会崩溃，属于有意设计。
 3. 测试缺口：本次未新增单测，覆盖依赖 `test_unified_radix_cache_kl_hybrid_bitexact` 与 `test_inkling*` 等 bit-exact 测试；若这些测试未覆盖特定 capture 形状，回归风险仍在。
 - 影响：对用户：Inkling 系列模型在开启 MTP/EAGLE 且启用 CUDA graph 时不再启动崩溃，服务可正常拉起。对系统：修复了 `multilayer_eagle_draft_extend_cuda_graph_runner` 与 `inkling_sconv_backend` 在捕获阶段的交互缺陷，消除了 mamba 状态缓存路径上的一个启动期故障。对团队：为 #34899（Unified Radix Cache bit-exact 正确性覆盖）中 MTP 项的 per-commit guard 扫清了阻塞条件。
- 风险标记：缺少测试覆盖，CUDA graph 捕获路径变更，核心路径变更

关联脉络

- PR #34043 [Referenced in PR body] Changed graph path to fill inert track buffers: 本 PR 的根因分析显示 #34043 让捕获路径继续启动 track scatter，首次使 `extend_prefix_lens` 的读取可达；作者明确说明不能回滚它来修复，因为它修复了真实的状态损坏。
- PR #33023 [Referenced in PR body] Introduced the `extend_prefix_lens` read: 引入了对 `forward_batch.extend_prefix_lens` 的读取，但当时因无 tracking metadata 的 batch 提前返回而不可达；本 PR 修复的是该读取首次可达后的崩溃。