PR #23552 完整报告

sgl-project/sglang

Pre-set SWA cache location in CudaGraphRunner

合并时间：2026-04-24 07:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23552

执行摘要

- 一句话：CUDA 图捕获前预置 SWA 缓存位置，避免回退到逐层翻译路径。
- 推荐动作：值得精读。PR 展示了如何在 CUDA 图捕获中通过预置缓冲区强制走快速路径的典型手法，对理解 CUDA 图捕获与 KV 缓存交互有参考价值。建议关注后续的类型清理 PR。

功能与动机

对于混合 SWA 模型（如 Gemma），原有 CUDA 图捕获时 out_cache_loc_swa 未预置，导致 set_kv_buffer 在回放阶段会回退到每层调用 translate_loc_from_full_to_swa 的慢路径。本 PR 通过预分配张量并在图捕获前设置，强制捕获快速 GPU 路径，提升解码性能。

实现拆解

1. 新增数据结构字段：在 DecodeInputBuffers 数据类中添加 out_cache_loc_swa: Optional[torch.Tensor] 字段（文件 cuda_graph_runner.py:141），类型为 torch.int64（注意：review 指出更合理应为 int32，作者承诺后续清理）。
2. 条件分配张量：在 DecodeInputBuffers.create() 方法中增加 is_hybrid_swa: bool = False 参数，当该标志为 True 时分配 out_cache_loc_swa 张量（形状 (max_num_token,)，类型 int64），否则为 None（cuda_graph_runner.py:183-187）。该参数通过 CudaGraphRunner.__init__ 从 model_runner.is_hybrid_swa 传入。
3. 填充数据：在 populate_from_forward_batch() 方法中，当 self.out_cache_loc_swa 和 forward_batch.out_cache_loc_swa 均不为 None 时，将 forward_batch 中的 SWA 缓存位置数据拷贝到缓冲区中，与已有的 GPU 张量拷贝批次合并，利用 _grouped_foreach_copy_ 按 dtype 分组统一拷贝（cuda_graph_runner.py:364-371）。
4. 图捕获前预置：在 run_once() 函数捕获 CUDA 图之前，检查 self.buffers.out_cache_loc_swa 是否非 None，若是则调用 self.model_runner.token_to_kv_pool.set_swa_loc() 将该缓冲区地址注册到 KV 池中（cuda_graph_runner.py:1148-1156）。这确保图捕获时 set_kv_buffer 内部的 if self.swa_loc is not None 分支走快速 GPU 操作，而非运行时逐层查表翻译。

关键文件：

- python/sglang/srt/model_executor/cuda_graph_runner.py（模块 CUDA 图执行器；类别 source；类型 data-contract；符号 DecodeInputBuffers, DecodeInputBuffers.create, DecodeInputBuffers.populate_from_forward_batch, CudaGraphRunner.init）：唯一修

改的文件，包含所有变更：新增数据类字段、条件分配、数据填充、图捕获前置逻辑。

关键符号：DecodeInputBuffers.create, DecodeInputBuffers.populate_from_forward_batch, CudaGraphRunner.init, CudaGraphRunner.run_once

关键源码片段

python/sglang/srt/model_executor/cuda_graph_runner.py

唯一修改的文件，包含所有变更：新增数据类字段、条件分配、数据填充、图捕获前置逻辑。

```
# cuda_graph_runner.py 关键变更片段

@dataclass
class DecodeInputBuffers(ForwardInputBuffers):
    # ...
    out_cache_loc: torch.Tensor
    out_cache_loc_swa: Optional[torch.Tensor] # 新增：SWA 位置缓存，可选
    # ...

    @classmethod
    def create(
        cls,
        # ...
        is_hybrid_swa: bool = False, # 新增参数：是否混合 SWA 模型
    ) -> "DecodeInputBuffers":
        with torch.device(device):
            # ...
            out_cache_loc_swa = (
                torch.zeros((max_num_token,), dtype=torch.int64) # 注意：review 建议改为 int32
                if is_hybrid_swa
                else None
            )
            # ...
        return cls(
            # ...
            out_cache_loc_swa=out_cache_loc_swa,
        )

    def populate_from_forward_batch(self, *, forward_batch, ...):
        # ... 已有 GPU 拷贝逻辑后
        # SWA cache location (int32, separate from the int64 batch above)
        if (
            self.out_cache_loc_swa is not None
            and forward_batch.out_cache_loc_swa is not None
        ):
            dsts.append(self.out_cache_loc_swa[:raw_num_token])
            srcs.append(forward_batch.out_cache_loc_swa[:raw_num_token])
        # 与已有拷贝合并同 dtype 批次
        _grouped_foreach_copy_(dsts, srcs)
```

```
def run_once(self):
    # ... 捕获前设置 SWA 位置
    if self.buffers.out_cache_loc_swa is not None:
        self.model_runner.token_to_kv_pool.set_swa_loc(
            self.buffers.out_cache_loc_swa[:num_tokens]
        )
    # 之后开始图捕获
```

评论区精华

> 主要讨论：review 评论指出 out_cache_loc_swa 使用了 torch.int64 类型，但 SWA 缓存位置在池内部使用 int32，因此应该为 int32 以节省内存并保持一致性。作者（merrymercy）承认这是好发现，并指出代码库中多处混合使用了 int32/int64，计划在后续 PR 中统一清理。

- out_cache_loc_swa 张量类型应为 int32 而非 int64 (correctness): 作者承认问题，并计划在后续 PR 中统一清理代码库中 int32/int64 的混合使用。当前 PR 保留 int64 以尽快合并。

风险与影响

- 风险：
  1. 类型不匹配风险：新引入的 out_cache_loc_swa 使用 int64 类型，但 KV 池内部使用 int32，可能导致隐式类型转换或占用额外显存。作者已认可并计划后续修复。
  2. 缺少测试覆盖：改动涉及 CUDA 图捕获的关键路径，但未包含对应测试或回归验证，存在引入回归的隐患。
  3. 仅影响混合 SWA 模型：is_hybrid_swa 标志默认 False，对其他模型无影响，但新字段在非 SWA 场景下为 None，需检查所有调用点是否兼容可选字段。 - 影响：影响范围：影响使用混合 SWA 模型的 CUDA 图解码路径（如 Gemma），预计可减少每层翻译的 Python 开销，提升解码吞吐。影响程度：中等。性能优化针对特定模型家族，改动集中且无回归风险（新字段为可选）。团队影响：为后续 SWA 相关优化铺平道路，且作者已计划对 int32/int64 类型做统一清理。
- 风险标记：类型不匹配 (int64 vs int32)，缺少测试覆盖，仅影响混合 SWA 模型

关联脉络

- PR #23545 Fix MoE no_combine: skip router weight in down projection: 同属对 CUDA 图捕获路径的优化，涉及 KV 缓存操作。
- PR #23588 [PD+DP] Allow PrefillDelayer in disaggregated-prefill mode: 同样涉及调度与缓存位置管理，与本 PR 的 SWA 位置预置有间接关联。