PR #25922 完整报告

sgl-project/sglang

[core] Unify output_tokens_buf in FutureMap

合并时间：2026-05-22 04:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25922

执行摘要

- 一句话：统一 FutureMap 中的 output_tokens_buf，简化 spec 中继缓冲区
- 推荐动作：建议审核并合并。该 PR 是 FutureMap 重构系列的一部分，虽包含多处核心路径变更，但改动集中且经过 CI 验证。值得关注的设计决策是统一缓冲区布局以简化未来扩展。

功能与动机

根据 PR 描述，重构 FutureMap 以通过 dataclass 模式声明 spec v2 中继缓冲区，从而简化添加新中继字段的过程。实际变更为统一缓冲区布局并清理 API，降低维护成本。

实现拆解

1. 统一 output_tokens_buf：在 overlap_utils.py 的 FutureMap.__init__ 中，将原本非 spec 模式使用的 token_ids_buf 和 spec 模式使用的 bonus_tokens_buf 合并为一个 output_tokens_buf。bonus_tokens_buf 的懒分配逻辑被移除，bonus_tokens 直接读取 output_tokens_buf。
2. 清理发布 - 暂存机制：将 on_verify_complete 参数重命名为 on_publish，并统一两处 speculative worker 的调用签名。publish_ready 类型从 Optional[torch.cuda.Event] 改为设备无关的惰性初始化，移除对 CUDA 的硬依赖。
3. 简化 scheduler 调用：在 scheduler.py 的 run_batch 中，移除了对 batch.is_spec_v2 的守卫，直接调用 self.future_map.resolve_seq_lens_cpu(batch)，该方法内部会通过 spec_info.future_indices 自检是否有效。
4. 增加断言保护：在 resolve_future 方法中，添加 torch._assert_async((batch.seq_lens > 0).all()) 用于异步检查 seq_lens 是否为正，帮助快速捕获逻辑错误。
5. 配套修改：更新了 eagle_worker_v2.py 和 multi_layer_eagle_worker_v2.py 中 forward_batch_generation 方法的参数名，以及 scheduler.py 中的调用点。所有变更不引入新测试，依赖现有 CI 覆盖。

关键文件：

- python/sglang/srt/managers/overlap_utils.py（模块 调度器；类别 source；类型 dependency-wiring；符号 stash, publish, resolve_future, resolve_seq_lens_cpu）：核心变更文件，实现了 output_tokens_buf 的统一、publish/stash 逻辑调整及设备无关性改进。

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：修改了 run_batch 中的调用模式：移除 is_spec_v2 守卫，直接调用 resolve_seq_lens_cpu；参数重命名。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 forward_batch_generation）：参数重命名：forward_batch_generation 方法中 on_verify_complete -> on_publish。
- python/sglang/srt/speculative/multi_layer_eagle_worker_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 forward_batch_generation）：与 eagle_worker_v2.py 相同的参数重命名变更。

关键符号：FutureMap.init, FutureMap.stash, FutureMap.publish, FutureMap.resolve_future, FutureMap.resolve_seq_lens_cpu, EagleWorkerV2.forward_batch_generation, MultiLayerEagleWorkerV2.forward_batch_generation, Scheduler.run_batch

关键源码片段

python/sglang/srt/managers/overlap_utils.py

核心变更文件，实现了 output_tokens_buf 的统一、publish/stash 逻辑调整及设备无关性改进。

```
# python/sglang/srt/managers/overlap_utils.py

class FutureMap:
    def __init__(self, device, spec_algo, req_to_token_pool):
        self.device = device
        self.spec_algo = spec_algo
        self.req_pool_size = req_to_token_pool.req_to_token.shape[0]

        # 统一的 output_tokens_buf: 非 spec 模式下存储 next_token_ids,
        # spec 模式下存储 bonus_tokens，减少缓冲区种类。
        self.output_tokens_buf = torch.empty(
            (self.req_pool_size,), dtype=torch.int64, device=self.device
        )
        if not self.spec_algo.is_none():
            self.new_seq_lens_buf = torch.empty(
                (self.req_pool_size,), dtype=torch.int64, device=self.device
            )
            self._forward_buf_initialized = False
        # publish_ready 初始化为 None，首次 publish 时惰性创建 Event,
        # 避免对 cuda.Event 的硬编码，兼容 HIP。
        self.publish_ready = None

    def stash(self, fi: FutureIndices, payload: Union[torch.Tensor, EagleDraftInput]):
        # stash 语义：将 forward stream 中的计算结果写入 output_tokens_buf
        # 或 new_seq_lens_buf，使用 index_copy_ 确保流序。
        if isinstance(payload, EagleDraftInput): # spec v2
```

```
            self.new_seq_lens_buf.index_copy_(0, fi.indices, payload.new_seq_lens.to(device=self.
            device, non_blocking=True))
            self.output_tokens_buf.index_copy_(0, fi.indices, payload.bonus_tokens.to(device=self.
            device, non_blocking=True))
        else: # non-spec
            self.output_tokens_buf.index_copy_(0, fi.indices, payload.to(device=self.device, non_
            blocking=True))

    def resolve_future(self, batch: ScheduleBatch):
        if self.spec_algo.is_none():
            # 非 spec 模式：从 output_tokens_buf 解析被占位的 token ids
            _resolve_future_token_ids(batch.input_ids, self.output_tokens_buf)
        else:
            draft_input = batch.spec_info
            if draft_input is None:
                return
            indices = draft_input.future_indices.indices
            # 从 output_tokens_buf 读取 bonus_tokens
            draft_input.bonus_tokens = self.output_tokens_buf[indices]
            draft_input.new_seq_lens = self.new_seq_lens_buf[indices]
            batch.seq_lens = draft_input.new_seq_lens
            # 异步断言：确认 seq_lens 大于 0，否则可能是 fence 错误
            torch._assert_async((batch.seq_lens > 0).all())
            if spec_need_hidden_states():
                draft_input.hidden_states = self.hidden_states_buf[indices]
```

评论区精华

该 PR 无公开 review 评论，作者自行合并。CI 运行通过，无讨论争议。

- 暂无高价值评论线程

风险与影响

- 风险：风险集中在核心调度和推测解码路径：
  - output_tokens_buf 的统一确保 int64 类型，兼容两种模式，但若未来引入其他类型需注意。
  - publish_ready 的惰性创建与设备解耦，但首次使用时的行为保持不变，风险低。
  - 参数重命名 on_verify_complete -> on_publish 需要确保所有调用点已更新，代码检查已覆盖。
  - 新增的 assert_async 不会影响性能，但可能在意外条件下触发，需要验证。
  - 移除 is_spec_v2 守卫使 resolve_seq_lens_cpu 对所有批次都调用，但内部自检会快速返回，无实质影响。
  - 影响：对最终用户无直接可见影响。对开发者而言，FutureMap 的接口更加清晰，缓冲区管理更一致。对系统维护性有正面影响，并为进一步的 speculative decoding 重构铺平道路。

- 风险标记：核心路径变更，参数重命名需同步，缓冲区类型假设

关联脉络

- PR #25879 [Spec] Route seq_lens through FutureMap; drop verify_done.wait: 本 PR 是 25879 的延续，进一步完善 FutureMap 的缓冲区布局和 API 命名。
- PR #25962 [Spec] Polish FutureMap after #25879: rename callback, async guard, cleanup: 同期 FutureMap 清理 PR，与本 PR 目标重叠，缓冲区统一部分在此 PR 中最终落地。