PR #25944 完整报告

sgl-project/sglang

[core] step 1: route non-spec `seq_lens` via `FutureMap` with per-mode bootstrap fixes

合并时间：2026-05-22 11:15

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25944

执行摘要

- 一句话：路由非推测 seq_lens 至 FutureMap，统一跨模式处理
- 推荐动作：建议阅读以理解 FutureMap 在调度中的统一设计，特别是 invalidate 和 resolve_future 的对称关系。配合下一步 #26020 可了解完整演进方向。

功能与动机

受 #25922 启发，跨推测 / 非推测重叠模式之间 seq_lens 处理存在重复逻辑和跨流竞争风险。非推测模式下原在 prepare_for_decode 中通过 batch.seq_lens = batch.seq_lens + 1 更新 GPU 张量，与 spec_v2 已通过 FutureMap 中继的方式不一致。本 PR 旨在统一两者，提高正确性和可维护性，并为后续消除 sentinel 机制铺垫（#26020）。

实现拆解

1. 废弃 EagleDraftInput.new_seq_lens：将该字段移至 GenerationBatchResult.new_seq_lens，消除在 spec 和非 spec 间通过不同路径传递的冗余，并简化构造函数。
2. FutureMap 统一 new_seq_lens_buf：原先只为 spec 模式创建 new_seq_lens_buf，现默认创建；resolve_future 中新增 is_decode() 分支，为非推测 decode 从 buf 恢复 batch.seq_lens，并增加 _resolve_spec_extras 方法分离 spec 专属逻辑。
3. 调度器 publish 和 invalidate：在 scheduler.run_batch 的 overlap 分支中，非推测模式下在 forward 后直接调用 future_map.publish(future_indices, batch.seq_lens + 1)；新增 future_map.invalidate(batch, future_indices) 统一将 batch.input_ids 和 batch.seq_lens 设为 sentinel（-future_indices.indices），替换原先仅设置 input_ids 的分散逻辑。
4. 修复 sentinel 副作用：在 alloc_for_decode 中检测 enable_overlap 后从 batch.seq_lens_cpu 物化临时 GPU 张量，避免分配器读取 sentinel 导致 KV 槽错误；在 mix_with_running 中先恢复 running_batch.seq_lens 再 merge，防止 MIXED 批次传播 sentinel。
5. 非 spec 解聚合 PREBUILT 启动：在 process_prebuilt 中，为非推测重叠模式添加 FutureMap 启动代码（publish + stash），使得首轮 DECODE 可正常从 buf 解析 seq_lens。

关键文件：

- python/sglang/srt/managers/overlap_utils.py（模块 调度器；类别 source；类型 core-logic；符号 _resolve_spec_extras, invalidate）：核心变更，FutureMap 类重构：新增 _resolve_spec_extras 和 invalidate，resolve_future 统一处理 decode 非推测和推测分支
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 run_batch）：调度器 run_batch 中统一调用 publish 和 invalidate，移除显式 sentinel 赋值
- python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py（模块 解聚合；类别 source；类型 dependency-wiring；符号 process_prebuilt）：为 disagg 非推测 PREBUILT 模式添加 FutureMap 启动 (publish + stash)，修复首轮 decode 未初始化问题
- python/sglang/srt/mem_cache/common.py（模块 缓存层；类别 source；类型 core-logic；符号 alloc_for_decode）：alloc_for_decode 中新增 enable_overlap 分支，从 seq_lens_cpu 物化 GPU 张量，避免读取 sentinel
- python/sglang/srt/managers/schedule_batch.py（模块 调度批次；类别 source；类型 core-logic；符号 prepare_for_decode, mix_with_running）：prepare_for_decode 移除重叠模式的 GPU += 1，改为仅更新 CPU shadow；mix_with_running 恢复 seq_lens 避免 sentinel 传播

关键符号：FutureMap.resolve_future, FutureMap._resolve_spec_extras, FutureMap.invalidate, FutureMap.publish, ScheduleBatch.prepare_for_decode, ScheduleBatch.mix_with_running, alloc_for_decode, process_prebuilt

关键源码片段

python/sglang/srt/managers/overlap_utils.py

核心变更，FutureMap 类重构：新增 _resolve_spec_extras 和 invalidate，resolve_future 统一处理 decode 非推测和推测分支

```
class FutureMap:
    def resolve_future(self, batch: ScheduleBatch):
        # 非推测 decode 从 buf 恢复 seq_lens；推测 decode 由 _resolve_spec_extras 处理
        if batch.forward_mode.is_decode():
            batch.seq_lens = self.new_seq_lens_buf[batch.req_pool_indices]
            torch._assert_async((batch.seq_lens > 0).all())

        if self.spec_algo.is_none():
            _resolve_future_token_ids(batch.input_ids, self.output_tokens_buf)
        else:
            self._resolve_spec_extras(batch)

    def _resolve_spec_extras(self, batch: ScheduleBatch) -> None:
        draft_input: EagleDraftInput = batch.spec_info
        if draft_input is None:
            # FIXME(lsyin): only prefill; not compatible with mixed mode
            return
        indices = draft_input.future_indices.indices
```

```
        indices.record_stream(torch.get_device_module(self.device).current_stream())
        draft_input.topk_p = self.topk_p_buf[indices]
        draft_input.topk_index = self.topk_index_buf[indices]
        draft_input.bonus_tokens = self.output_tokens_buf[indices]
        if spec_need_hidden_states():
            draft_input.hidden_states = self.hidden_states_buf[indices]
        # 注意：不再原地恢复 seq_lens，统一在 resolve_future 中处理

    def invalidate(self, batch: ScheduleBatch, future_indices: FutureIndices) -> None:
        # 在两次 forward 之间设置 sentinel：-indices
        sentinel = -future_indices.indices
        batch.input_ids = sentinel
        batch.seq_lens = sentinel

    def publish(self, future_indices: FutureIndices, new_seq_lens: torch.Tensor) -> None:
        indices = future_indices.indices
        if indices.shape[0] == 0:
            return # DP idle
        self.new_seq_lens_buf[indices] = new_seq_lens.to(self.new_seq_lens_buf.dtype)
        if self.spec_algo.is_some():
            if self.publish_ready is None:
                self.publish_ready = torch.get_device_module(self.device).Event()
            self.publish_ready.record()
```

python/sglang/srt/managers/scheduler.py

调度器 run_batch 中统一调用 publish 和 invalidate，移除显式 sentinel 赋值

```
  def run_batch(self, batch, pp_proxy_tensors=None):
      ...
      if self.enable_overlap:
          self.future_map.resolve_seq_lens_cpu(batch)
          with self._overlap_forward_isolation(batch):
              future_indices = FutureIndices(indices=batch.req_pool_indices)
              fwd_kwargs = (
                  {"on_publish": partial(self.future_map.publish, future_indices)}
                  if batch.is_spec_v2
                  else {}
              )
              with self.forward_stream_ctx:
                  self.forward_stream.wait_stream(self.schedule_stream)
                  self.future_map.resolve_future(batch)
                  batch_result = self.model_worker.forward_batch_generation(batch, **fwd_kwargs)
                  # 非推测模式：调度器在 forward 后直接 publish，无需 on_publish 回调
                  if not batch.is_spec_v2:
                      self.future_map.publish(future_indices, batch.seq_lens + 1)
                  # 保持引用等处理 ...
              # 统一 invalidate，同时设置 input_ids 和 seq_lens 为 sentinel
              self.future_map.invalidate(batch, future_indices)
              if batch.is_spec_v2:
```

```
            batch.spec_info = batch_result.next_draft_input
            batch.spec_info.future_indices = future_indices
    ...
```

评论区精华

本 PR 未产生公开 review 讨论；变更由作者独立提交并合并。

- 暂无高价值评论线程

风险与影响

- 风险：修改集中在核心调度路径（scheduler、overlap_utils、schedule_batch、mem_cache），引入 sentinel 和 FutureMap 统一后，若 publish 和 resolve 顺序错乱或 buf 索引错误，可能导致 batch.seq_lens 读入无效值，进而引发 KV 分配错误或其他隐式错误（但已添加 _assert_async 检测）。disagg 路径的启动改动较新，可能与其他未覆盖的 prebuilt 模式冲突。当前没有直接对应的单元测试，但 CI extra 已通过。
- 影响：对用户无直接功能变化，但改善了内部一致性和可靠性。影响所有启用 overlap 的推理场景（推测和非推测），包括 disaggregation 下的 decode 事件循环。影响程度中等，因为涉及基础架构重构，但通过 CI 验证。
- 风险标记：核心路径变更，跨流同步依赖，跨模式兼容修复，无测试文件配套

关联脉络

- PR #25922 [core] Unify output_tokens_buf in FutureMap: 前置 PR，引入 FutureMap 并统一 output_tokens_buf；本 PR 在此基础上统一 seq_lens 处理
- PR #26020 [core] step 2: drop sentinel mechanism (not in history but mentioned in PR body): 下一阶段将废除 sentinel 机制，使本 PR 中的部分修复变为冗余