PR #26020 完整报告

sgl-project/sglang

[core] step 2: drop seq_lens sentinel; SB maintains GPU as `seq_lens_cpu` mirror

合并时间：2026-05-22 15:12

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26020

执行摘要

- 一句话：丢弃 seq_lens sentinel，统一 GPU/CPU 镜像维护
- 推荐动作：该 PR 值得精读，展示了如何将分散的临时修复整合为统一不变量的重构方法。关键设计决策：单一入口 ForwardBatch.init_new 作为 GPU materialization 点；SB 只维护镜像，前向路径只写 forward_batch。后续开发应参考此模式，避免再次出现分散协调点。

功能与动机

前一步 #25944 留下了 mode mix 问题：SB.seq_lens GPU 的有效性依赖于 batch 处于哪种模式，协调逻辑分散在 5 个地方（non-overlap、overlap + spec_v2、overlap + non-spec、mixed、alloc_for_decode）。本 PR 通过建立干净的不变量统一处理，消除维护负担和潜在错误。

实现拆解

1. 重构 FutureMap 机制：将 invalidate 方法拆分为 set_input_ids_sentinel（只设置 input_ids sentinel），不再设置 seq_lens sentinel；修改 resolve_seq_lens_cpu 在拉取 CPU 值的同时更新 GPU 镜像 (batch.seq_lens = new_seq_lens)，保证 SB.seq_lens 始终与 CPU 一致。
2. 统一 SB 维护 GPU/CPU 镜像：在 ScheduleBatch.prepare_for_decode 中，overlap 模式不再跳过 GPU 直接 add_，而是通过 self.seq_lens = self.seq_lens + 1 新建 tensor，保持 non-overlap 和 overlap 路径都维护镜像一致。
3. 移除分散的 fallback 修复：删除了 mix_with_running 中从 CPU 恢复 GPU seq_lens 的代码、alloc_for_decode 中从 CPU materialize 的 overlap 分支、以及 disagg non-spec PREBUILT 中对 FutureMap 的 bootstrap 调用。
4. 调整 spec_v2 的 seq_lens 突变位置：将 EagleDraftInputV2Mixin.prepare_for_extend_to_fill_draft_kvcache 中对 batch.seq_lens 的直接修改移到 forward_batch 上，避免污染 SB 镜像。
5. 调度器调用更新：将 run_batch 中的 invalidate 调用改为 set_input_ids_sentinel。

涉及文件：overlap_utils.py、schedule_batch.py、mem_cache/common.py、decode_schedule_batch_mixin.py、eagle_info_v2.py、scheduler.py。

关键文件：

- python/sglang/srt/managers/overlap_utils.py（模块 重叠调度；类别 source；类型 core-logic；符号 invalidate, set_input_ids_sentinel, resolve_seq_lens_cpu, resolve_future）：核心变更：FutureMap.invalidate 拆分为 set_input_ids_sentinel, resolve_seq_lens_cpu 同时更新 GPU 镜像，resolve_future 不再恢复 seq_lens，是统一镜像的关键。
- python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py（模块 分离部署；类别 source；类型 dependency-wiring）：移除了 non-spec PREBUILT 路径中对 FutureMap 的 bootstrap 调用（publish + stash），因为 SB 自身维护 seq_lens 镜像，不再需要提前发布。
- python/sglang/srt/managers/schedule_batch.py（模块 调度批处理；类别 source；类型 core-logic）：prepare_for_decode 中 overlap 模式改为新建 tensor＋1（替换原跳过 in-place add），mix_with_running 移除 GPU 恢复代码，确保 GPU/CPU 镜像一致。
- python/sglang/srt/mem_cache/common.py（模块 内存缓存；类别 source；类型 core-logic）：alloc_for_decode 移除了 overlap 分支中用 seq_lens_cpu.to(device) 的 fallback，直接使用 batch.seq_lens（新不变量保证其正确）。
- python/sglang/srt/speculative/eagle_info_v2.py（模块 推测解码；类别 source；类型 core-logic）：将 spec_v2 中 draft extend 对 seq_lens 的修改从 batch 移到 forward_batch，避免污染 SB 镜像。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：将 invalidate 调用改为 set_input_ids_sentinel，反映方法重命名和语义变化。

关键符号：FutureMap.set_input_ids_sentinel, FutureMap.resolve_seq_lens_cpu, FutureMap.resolve_future, ScheduleBatch.prepare_for_decode, ScheduleBatch.mix_with_running, alloc_for_decode, EagleDraftInputV2Mixin.prepare_for_extend_to_fill_draft_kvcache

关键源码片段

python/sglang/srt/managers/overlap_utils.py

核心变更：FutureMap.invalidate 拆分为 set_input_ids_sentinel，resolve_seq_lens_cpu 同时更新 GPU 镜像，resolve_future 不再恢复 seq_lens，是统一镜像的关键。

```
class FutureMap:
    """Cross-iter relay buffer for values the next iter's schedule cannot
    compute locally (e.g. spec_v2 seq_lens after accept_lens, sampled tokens).

    Forward stream publishes into a buf; next iter's schedule pulls lazily.
    Schedule-deterministic values (e.g. non-spec seq_lens via +1) stay
    maintained by SB directly and do not need the relay.

    SB.seq_lens GPU is always a faithful seq_lens_cpu mirror; forward path
    treats it as read-only, spec mutations land on forward_batch.seq_lens.
    """

    def set_input_ids_sentinel(
```

```
        self, batch: ScheduleBatch, future_indices: FutureIndices
    ) -> None:
        # 只为 input_ids 设置 sentinel（负数索引），不再设置 seq_lens sentinel。
        # resolve_future 通过 output_tokens_buf 将负数转换回实际 token。
        batch.input_ids = -future_indices.indices

    def resolve_seq_lens_cpu(self, batch: ScheduleBatch) -> None:
        # 从 new_seq_lens_buf 拉取 spec_v2 的 seq_lens，同时写入 GPU 和 CPU，
        # 保持 SB.seq_lens 与 seq_lens_cpu 镜像一致。
        fi = batch.spec_info.future_indices if batch.spec_info is not None else None
        if fi is None:
            return
        if self.publish_ready is not None:
            self.publish_ready.wait()
        new_seq_lens = self.new_seq_lens_buf[fi.indices]
        batch.seq_lens = new_seq_lens # 更新 GPU 镜像
        batch.seq_lens_cpu = new_seq_lens.cpu() # 同步 CPU
        batch.seq_lens_sum = int(batch.seq_lens_cpu.sum())

    def resolve_future(self, batch: ScheduleBatch):
        # 现在只解析 token ids 和 spec extras，不再解析 seq_lens，
        # 因为 SB.seq_lens 在进入此函数时已经是真实值。
        if self.spec_algo.is_none():
            _resolve_future_token_ids(batch.input_ids, self.output_tokens_buf)
        else:
            self._resolve_spec_extras(batch)
```

注意：resolve_future 中的 _resolve_spec_extras 用于解析 topk_p、topk_index、bonus_tokens、hidden_states 等 speculation 专用数据。

评论区精华

该 PR 没有收到任何 review 评论，所有决策由作者 hnyls2002 独立完成。主要设计讨论体现在 PR 描述和 16 次 commit 的演进中，包括从最初依赖 FutureMap 到最终统一镜像的逐步收敛。

- 暂无高价值评论线程

风险与影响

- 风险：新不变量依赖所有路径正确维护 SB.seq_lens GPU 镜像。如果某条路径意外修改了 batch.seq_lens 而未同步 CPU，或依赖旧 sentinel 行为，可能导致分配错误或解码失败。spec_v2 路径的修改（从 batch 移到 forward_batch）需要确保所有使用 forward_batch.seq_lens 的地方都已覆盖。此外，overlap 模式下 seq_lens = seq_lens + 1 新建 tensor 会略微增加内存分配开销，但消除了跨 stream 竞争。目前没有新增测试覆盖这些重构后的场景，回归风险较高。
- 影响：对用户：无直接影响，内部重构。对系统：统一 seq_lens 处理减少条件分支和错误根源，降低未来维护成本；为后续拆分 relay variables 和 transient variables 奠定基础。对

团队：需要确保所有新代码遵循新不变量，现有测试应覆盖主要场景，但缺少专门的回归测试。

- 风险标记：跨模式回归风险，缺少测试覆盖，overlap 路径变更，spec_v2 突变位置调整

关联脉络

- PR #25944 [core] step 1: route non-spec seq_lens via FutureMap with per-mode bootstrap fixes: 本 PR 是 step 2，直接基于 #25944 的改动，统一其引入的 per-mode 修复。
- PR #25922 （未在历史列表中）：PR body 提到 follow up on #25922，作为更早期的基础。