PR #30435 完整报告

sgl-project/sglang

[Fix] Chain the seq_lens publish event records so prebuilt seeding keeps the forward fence

合并时间：2026-07-09 04:43

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30435

执行摘要

- 一句话：修复 publish event 链式记录，消除预填充撕裂窗口
- 推荐动作：值得精读，尤其关注 CUDA event 的链式记录设计模式。该模式可用于其他需要累积语义的事件同步场景。改动简洁、注释清晰，是典型的细粒度正确性修复。

功能与动机

PR body 指出：在 overlap 模式下，FutureMap.publish 将 new_seq_lens 写入 relay buf 并记录 publish_ready 事件，下一轮迭代的 resolve 等待该事件以排序跨流读取。PD-decode 预构建路径没有 forward 流，它会在 scheduler 的 schedule stream 上调用 publish。CUDA event 只能记录单个 capture point，这会使得 publish_ready 脱离 forward 流，丢掉 in-flight forward 的写入栅栏。虽然目前依赖调度器 WAR barrier 等隐式排序在 CUDA 上恰好正确，但在没有 WAR barrier 的平台上存在过时读窗口。

实现拆解

1. 改动 FutureMap.publish 中的事件记录逻辑（python/sglang/srt/managers/overlap_utils.py）：在条件分支中，当 publish_ready 已存在时，先让当前 stream wait_event(self.publish_ready)，再 record()，从而将事件串联成累积语义——事件触发意味着之前所有 publish 的写入均可见。
2. 更新 eagle_disaggregation.py 的注释（python/sglang/srt/speculative/eagle_disaggregation.py）：在调用 publish 处添加注释，说明预构建种子操作通过链式记录保持 in-flight forward 的栅栏完整。
3. 行为保持不变：forward 路径上的 publish 在同一 forward stream 上调用，链式 wait 是同流空操作，不会影响稳态解码性能。预构建种子路径（off forward stream）现在显式等待 in-flight forward 的 publish event，而非意外丢弃。

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 FutureMap.publish）：核心修改：在 FutureMap.publish 方法中增加了链式事件记录逻辑，将 publish_ready 事件从单次捕获升级为累积语义，修复跨流同步的竞态条件。
- python/sglang/srt/speculative/eagle_disaggregation.py（模块 推测解码；类别 source；类型 core-logic；符号 build_eagle_disagg_draft_input）：辅助变更：在构建 Eagle 解耦

draft 输入时调用 publish 处添加注释，说明链式记录保持 in-flight forward 栅栏完整。

关键符号：未识别

关键源码片段

python/sglang/srt/managers/overlap_utils.py

核心修改：在 FutureMap.publish 方法中增加了链式事件记录逻辑，将 publish_ready 事件从单次捕获升级为累积语义，修复跨流同步的竞态条件。

```
def publish(self, future_indices: torch.Tensor, new_seq_lens: torch.Tensor) -> None:
    indices = future_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)
    # Only spec_v2 needs the event; it gates the seq_lens D2H on the private stream.
    if self.spec_algo.is_some():
        device_module = torch.get_device_module(self.device)
        if self.publish_ready is None:
            self.publish_ready = device_module.Event()
        else:
            # Chain the records: event fire implies every prior publish is
            # visible, so an off-forward-stream publish (PD-decode prebuilt
            # seeding) cannot drop the in-flight forward's fence.
            device_module.current_stream().wait_event(self.publish_ready)
        self.publish_ready.record()
```

python/sglang/srt/speculative/eagle_disaggregation.py

辅助变更：在构建 Eagle 解耦 draft 输入时调用 publish 处添加注释，说明链式记录保持 in-flight forward 栅栏完整。

```
if batch.enable_overlap:
    spec_info.future_indices = batch.req_pool_indices
    # Seed the relay buf with the known seq_lens; publish's chained record
    # keeps the in-flight forward's fence intact (see FutureMap.publish).
    future_map.publish(spec_info.future_indices, batch.seq_lens)
    future_map.stash(
        spec_info.future_indices, RelayPayload.from_draft_input(spec_info)
    )
```

评论区精华

PR 无 review 评论讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：改动位于 spec_v2 专用的条件分支内，且是对现有事件的累积性增强（wait + record），而非删除或替代现有同步逻辑。CUDA 上行为保持兼容，仅预构建

batch 后紧接的迭代会真正等待 forward publish 完成，而非意外跳过等待。无 WARbarrier 的平台（如部分 AMD GPU）将从竞态修复中受益。代码行数少、逻辑自包含，回归风险小。

- 影响：
  - 用户：PD 分离部署场景下，spec_v2 用户的 *decode* 阶段不再可能读取陈旧 seq_lens，尤其是非 CUDA 平台上的稳定性提升。
  - 系统：无性能退化，forward 路径的链式 wait 是同流 no-op。
  - 团队：消除了靠三处无关代码隐式排序才能正确的脆弱依赖，降低维护负担。
  - 风险标记：遗留隐式依赖去除，缺少测试覆盖

关联脉络

- PR #29834 Fix scheduler crash on prefill-unreachable *decode* abort: 同为调度 / 解耦场景下的正确性修复，涉及跨流同步和竞态条件。
- PR #30348 [refactor] ctx.resources: named slots, stream leases, and workspace buffer leases: 涉及流资源管理的重构，与跨流同步机制有关联。