

PR #35748 完整报告

sgl-project/sglang

Fix overlap prebuilt row reuse race

合并时间: 2026-08-21 17:00

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35748>

执行摘要

- 一句话: 修复 overlap 预构建行复用竞态, 调度流等待前向流
- 推荐动作: 值得精读。该 PR 修复的是调度核心路径上的竞态条件, 涉及流同步和 FutureMap 的共享状态管理。理解其修复思路 (通过显式流等待代替事件链) 对处理类似并发问题有参考价值。此外, 其测试编写方式 (使用 mock 验证调用顺序) 也值得学习。

功能与动机

PR body 明确指出这是一个 FutureMap 中的竞态条件 bug: `publish()` 可能在两种情况下被调用: (1) 前向流在 `draft+verify` 之后; (2) 调度流中, 当 PD disagg 启用且 `prefill` 请求到达时。`publish()` 会修改 FutureMap 的共享状态, 因此必须正确排序以避免竞态。`publish_ready` 事件不能用于同步, 因为 `stash()` 在 `record` 之后也会修改 FutureMap 共享状态。Reviewer `merrymercy` 也确认这是实际存在的竞态: 'an actual race condition when an already finished forward is still running, while PD-decode prebuilt requests come in and occupy the same lane.'

实现拆解

实现分三步:

1. 核心修复 (`decode.py`): 在 `get_new_prebuilt_batch` 中, 当 `enable_overlap` 为真时, 在调用 `new_batch.process_prebuilt(self.future_map)` 之前, 先调用 `self.schedule_stream.wait_stream(self.forward_stream)`, 确保前向流中未完成的 `publish()` 操作已经完成, 避免复用同一请求池行时出现数据竞争。
2. 简化 `publish()` (`overlap_utils.py`): 移除了 `publish()` 中对 `publish_ready` 事件进行链式等待的逻辑 (`device_module.current_stream().wait_event(self.publish_ready)`), 因为现在发布操作已经通过流同步保证顺序, 不再需要此链式等待, 避免了潜在的事件记录顺序问题。
3. 测试配套 (`test_priority_scheduling_disaggregation.py`): 重构并新增测试。将原有 `TestDecodePrebuiltPriority` 测试类重构为 `TestDecodePrebuilt`, 提取 `_new_scheduler` 辅助方法, 并新增 `test_overlap_waits_for_forward_before_processing_prebuilt` 测试, 验证 `prepare_for_prebuilt`、`wait_stream`、`process_prebuilt` 的执行顺序。

关键文件:

- python/sglang/srt/disaggregation/decode.py (模块 调度器; 类别 source; 类型 core-logic; 符号 get_new_prebuilt_batch, process_decode_queue) : 核心修复位置: 在 get_new_prebuilt_batch 中, 当 enable_overlap 时, 在 process_prebuilt 前添加 wait_stream, 防止竞态。
- python/sglang/srt/managers/overlap_utils.py (模块 重叠调度; 类别 source; 类型 core-logic; 符号 FutureMap.publish, FutureMap.stash, FutureMap.prepare) : 简化 publish() 的链式事件记录, 避免潜在的问题。
- test/registered/unit/managers/test_priority_scheduling_disaggregation.py (模块 测试; 类别 test; 类型 test-coverage; 符号 TestDecodePrebuilt, _new_scheduler, test_overlap_waits_for_forward_before_processing_prebuilt) : 新增测试覆盖竞态修复, 验证执行顺序。

关键符号: get_new_prebuilt_batch, process_decode_queue, FutureMap.publish, FutureMap.stash, FutureMap.prepare

关键源码片段

python/sglang/srt/disaggregation/decode.py

核心修复位置: 在 get_new_prebuilt_batch 中, 当 enable_overlap 时, 在 process_prebuilt 前添加 wait_stream, 防止竞态。

```
# python/sglang/srt/disaggregation/decode.py
# 核心修复: 在 prebuilt 批处理前等待前向流, 防止竞态
def get_new_prebuilt_batch(self, running_batch):
    # ... 其他逻辑
    # 构造 fake completed prefill
    new_batch.prepare_for_prebuilt()
    if self.enable_overlap:
        # 已完成的请求可能仍有一个冗余前向在飞行中。
        # 在 prebuilt 请求复用可能被复用的请求池行之前, 先 drain 它。
        self.schedule_stream.wait_stream(self.forward_stream)
    new_batch.process_prebuilt(self.future_map)
    return new_batch
```

python/sglang/srt/managers/overlap_utils.py

简化 publish() 的链式事件记录, 避免潜在的问题。

```
# python/sglang/srt/managers/overlap_utils.py
# publish() 简化: 移除链式事件记录, 因为已通过流同步保证顺序
def publish(
    self,
    future_indices: torch.Tensor,
    new_seq_lens: torch.Tensor,
    confidence: Optional[torch.Tensor] = None,
) -> 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)
publish_confidence = self.needs_confidence_relay and confidence is not None
if publish_confidence:
    self.confidence_relay.scatter(indices, confidence)
# 只有 spec_v2 需要事件；它门控 seq_lens D2H 在私有流上。
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()
    self._publish_fresh = True
if publish_confidence:
    self.confidence_relay.issue_ring_copy(
        stream=self.fwd_prepare_d2h_stream,
        publish_ready=self.publish_ready,
    )

```

test/registered/unit/managers/test_priority_scheduling_disaggregation.py

新增测试覆盖竞态修复，验证执行顺序。

```

# test/registered/unit/managers/test_priority_scheduling_disaggregation.py
# 新增测试：验证 overlap 时等待前向流
def test_overlap_waits_for_forward_before_processing_prebuilt(self):
    scheduler = self._new_scheduler(enable_overlap=True)
    scheduler.waiting_queue = [MagicMock(rid="request")]

    call_order = []
    new_batch = MagicMock()
    new_batch.prepare_for_prebuilt.side_effect = lambda: call_order.append("prepare")
    scheduler.schedule_stream.wait_stream.side_effect = lambda _: call_order.append("wait")
    new_batch.process_prebuilt.side_effect = lambda _: call_order.append("process")

    with patch(
        "sglang.srt.disaggregation.decode.ScheduleBatch.init_new", return_value=new_batch,
    ), get_context().override_server_args(disaggregation_decode_enable_radix_cache=False):
        ret = SchedulerDisaggregationDecodeMixin.get_new_prebuilt_batch(
            scheduler, scheduler.running_batch
        )

    self.assertIs(ret, new_batch)
    scheduler.schedule_stream.wait_stream.assert_called_once_with(scheduler.forward_stream)
    self.assertEqual(call_order, ["prepare", "wait", "process"])

```

评论区精华

审查者 hnyls2002 指出：LGTM, thanks for the fix. my previous PR #30435 was trying to fix the similar PR, but I didn't consider the stashing data races between prebuilt batch and normal batches. 这表明此修复解决了之前未覆盖的 stash 数据竞争问题，与历史 PR #30435 相关。merrymercy 确认这是实际竞态，Jialin 的评论强调了竞态发生的场景。

- 竞态条件的确认 (correctness): 确认修复必要。
- 与历史修复的关联 (design): 本修复补充了 stash 竞态修复。

风险与影响

- 风险：风险较低，但需注意：
 - 仅影响启用 overlap 的 PD 分离场景：修复在 enable_overlap 条件下新增了流同步，可能带来轻微的性能开销，但这是必要开销，且只在 PD 分离时发生。
 - 依赖流等待的正确性：wait_stream 的调用假定 schedule_stream 和 forward_stream 均存在且有效，如果这些属性未正确初始化，可能会引发异常。但测试中已覆盖。
 - 移除链式记录逻辑：若未来有其他路径在前向流之外调用 publish()，且不经 get_new_prebuilt_batch 的同步，可能会导致事件顺序问题回归。需要确保所有调用路径都受保护。
 - 影响：影响范围限于启用 overlap 的 PD 分离部署（enable_overlap 且 disaggregation_decode_enable_radix_cache 等配置）。修复后，请求池行的复用不会出现竞态，提高了稳定性和正确性。对性能影响极小，仅增加一次流等待调用。测试覆盖了优先级调度和 overlap 场景，降低了回归风险。
 - 风险标记：核心路径变更，新增测试覆盖

关联脉络

- PR #30435 Unknown: 评审者提及之前的修复尝试，与本 PR 相关。