PR #26380 完整报告

sgl-project/sglang

[core] WAR barrier for overlap schedule buffer writes, without fwd occupancy cost

合并时间：2026-05-27 14:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26380

执行摘要

- 一句话：修复重叠调度数据竞争，解除前向占用损失
- 推荐动作：核心逻辑改动精炼，设计巧妙（用私有流避免屏障串行化），值得深入阅读。但 AMD 和 GB 回归表明跨平台适配不充分，建议后续添加数据竞争测试和跨平台性能基准后再逐步推广。

功能与动机

重叠调度中，schedule_stream 写入 req_to_token_pool、full_to_swa_index_mapping 等共享 GPU 缓冲区，而前向仍在读取，造成写后读竞争。在 DeepSeek V4 解耦模式下表现为 GSM8K 精度波动（~0.87），需通过屏障修复同时避免前向占用退化（之前方案 -4%）。

实现拆解

1. WAR 屏障：在 Scheduler.event_loop_overlap、Scheduler.event_loop_overlap_disagg_decode、Scheduler.event_loop_overlap_disagg_prefill 三个重叠循环顶部插入 schedule_stream.wait_stream(forward_stream)，确保本轮调度等待前向读完。
2. 解除屏障串行化：屏障将 schedule_stream 链到前向结束，两个主机同步若留在 schedule_stream 会阻塞到前向完成。 - H2D：EagleDraftInput.prepare_for_decode 中 cur_kv_lens_cpu.to(device=batch.device) 改为 non_blocking=True，避免隐式同步。 - D2H：OverlapBypassBuffer.resolve_seq_lens_cpu 中 seq_lens.cpu() 阻塞拷贝改为：在私有 fwd_prepare_d2h_stream 上等 publish_ready 事件后，以 non_blocking 复制到 pinned 内存，主机通过 req_pool_indices_cpu 索引选出。
3. 新增 pinned 缓冲与流：OverlapBypassBuffer.__init__ 中分配 new_seq_lens_cpu_pinned（pin_memory）和 fwd_prepare_d2h_stream，仅在 CUDA/HIP 上生效。

关键文件：

- python/sglang/srt/managers/overlap_utils.py（模块 重叠缓冲；类别 source；类型 core-logic；符号 init, resolve_seq_lens_cpu）：核心修改：添加 pinned 缓冲和私有 D2H 流，将 seq_lens_cpu 同步移出 schedule_stream，是关键设计实现。
- python/sglang/srt/speculative/eagle_info_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 prepare_for_decode）：H2D 转为 non_blocking，避免屏障串行化后主

机阻塞，是性能恢复的关键辅助修改。

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 event_loop_overlap）：在主重叠循环入口添加 WAR 屏障，是数据竞争修复的核心触发点。
- python/sglang/srt/disaggregation/decode.py（模块 解耦解码；类别 source；类型 core-logic；符号 event_loop_overlap_disagg_decode）：在解耦解码重叠循环添加 WAR 屏障，覆盖 disagg decode 路径。
- python/sglang/srt/disaggregation/prefill.py（模块 解耦预填充；类别 source；类型 core-logic；符号 event_loop_overlap_disagg_prefill）：在解耦预填充重叠循环添加 WAR 屏障，覆盖 disagg prefill 路径。

关键符号：Scheduler.event_loop_overlap, Scheduler.event_loop_overlap_disagg_decode, Scheduler.event_loop_overlap_disagg_prefill, OverlapBypassBuffer.init, OverlapBypassBuffer.resolve_seq_lens_cpu, EagleDraftInput.prepare_for_decode

关键源码片段

python/sglang/srt/managers/overlap_utils.py

核心修改：添加 pinned 缓冲和私有 D2H 流，将 seq_lens_cpu 同步移出 schedule_stream，是关键设计实现。

```
def resolve_seq_lens_cpu(self, batch: ScheduleBatch) -> None:
    # Lazy pull from new_seq_lens_buf for spec_v2 (accept_lens not known to
    # schedule). Write into both CPU and GPU so SB.seq_lens stays a faithful
    # seq_lens_cpu mirror.
    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()
    batch.seq_lens = self.new_seq_lens_buf[fi]

    if self.fwd_prepare_d2h_stream is None or self.publish_ready is None:
        # bootstrap / non-CUDA path: blocking CPU copy on schedule stream
        batch.seq_lens_cpu = batch.seq_lens.cpu()
        batch.seq_lens_sum = int(batch.seq_lens_cpu.sum())
        return

    # Offload D2H to private stream: wait for publish event,
    # then copy to pinned memory non_blocking.
    self.fwd_prepare_d2h_stream.wait_event(self.publish_ready)
    with torch.get_device_module(self.device).stream(self.fwd_prepare_d2h_stream):
        self.new_seq_lens_cpu_pinned.copy_(self.new_seq_lens_buf, non_blocking=True)
    self.fwd_prepare_d2h_stream.synchronize()
    # host-select via req_pool_indices_cpu (precomputed host mirror)
    batch.seq_lens_cpu = self.new_seq_lens_cpu_pinned[batch.req_pool_indices_cpu]
    batch.seq_lens_sum = int(batch.seq_lens_cpu.sum())
```

评论区精华

- AMD 回归报告（@bingxche）：PR 合入后 AMD MI35x 8-GPU 出现 test_deepseek_r1_mxfp4_8gpu 超时、test_priority_scheduling 断言失败、DSv4 非 MTP 性能下降 ~20%。作者 @hnyls2002 分析均为 HIP 特有，CUDA 无影响。
- GB200/GB300 性能回归（@nvpohanh）：在 GPT-OSS、Kimi、Llama3、DeepSeek、Nemotron 等模型上观察到性能回退，作者尚未回复。
- AMD 平台回归问题 (correctness): 作者表示会调查并建议临时回退。
- GB200/GB300 性能回归 (performance): 待进一步分析，可能与非阻塞拷贝或流竞争有关。

风险与影响

- 风险：
  - AMD/HIP 兼容性：私有流与 non_blocking 拷贝在 HIP 上行为可能不同，已观察到三种失败模式，需针对性修复。
  - 性能回归风险：GB200/GB300 出现系统性性能下降，可能与屏障串行化或流资源竞争有关。
  - 缺少直接测试：变更未配套数据竞争或性能回归的自动化测试，回归需靠手动验证。
  - 流资源影响：新增 fwd_prepare_d2h_stream 在设备流数量紧张时可能竞争。
  - 影响：影响所有使用重叠调度（event_loop_overlap）的场景，包括普通解碼和 DeepSeek V4 解耦路径。正面：修复数据竞争，提高精度稳定性。负面：在 AMD 和部分 NVIDIA 平台上引入性能回退。团队需关注跨平台一致性测试和性能基准。
  - 风险标记：核心路径变更，跨平台兼容风险，AMD 平台回归，GB200/GB300 性能回归，缺少直接测试

关联脉络

- PR #26376 bidirectional wait_stream barrier: 被本 PR 替代，其对称屏障导致 -4% 前向占用
- PR #26425 req_pool_indices_cpu host mirror: 本 PR 依赖其引入 req_pool_indices_cpu，实现主机侧索引选择