

PR #29407 完整报告

sgl-project/sglang

Localize cur_batch field in Scheduler to avoid field-based state access

合并时间: 2026-07-10 08:55

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29407>

执行摘要

- 一句话: 调度器 cur_batch 字段局部化, 仅看门狗使用
- 推荐动作: 值得精读, 展示了逐步消除全局状态、局部化依赖的典型模式, 适合调度器等核心模块重构参考。

功能与动机

PR body 指出: self.cur_batch 是调度器范围属性, 被每个事件循环写入, 但唯一跨线程读者是看门狗, 用作调试信号。所有循环内消费者应使用局部变量而非字段, 以减少状态耦合。

实现拆解

1. 字段重命名: 在 scheduler.py 的 __init__ 和 init_running_status 中将 self.cur_batch 改为 self.cur_batch_for_debug。
2. 事件循环改造: 在 event_loop_normal、event_loop_overlap、PP 事件循环、disaggregation 事件循环中, 将 self.cur_batch = batch 改为 self.cur_batch_for_debug = batch, 并保留局部变量 cur_batch 用于后续消费。
3. 函数参数化: launch_batch_sample_if_needed 增加 cur_batch: ScheduleBatch 参数, 调用方传入局部变量; _pp_launch_batch 签名增加 cur_batch 参数。
4. 看门狗适配: 在 invariant_checker.py 的 create_scheduler_watchdog 中, 将 is_active 和 dump_info 中的 scheduler.cur_batch 改为 scheduler.cur_batch_for_debug。
5. MLX 与 disaggregation 同步: 更新 mlx/scheduler_mixin.py、disaggregation/decode.py、disaggregation/prefill.py 中的对应赋值。
6. 测试适配: 修改四个测试文件中对 cur_batch 的直接引用, 改为 cur_batch_for_debug。

关键文件:

- python/sglang/srt/managers/scheduler_pp_mixin.py (模块 调度器; 类别 source; 类型 core-logic; 符号 event_loop_pp, event_loop_pp_disagg_prefill, event_loop_pp_disagg_decode, _pp_launch_batch): PP 主循环的核心改造: 引入局部 cur_batch, 镜像到 cur_batch_for_debug, 并传参给 _pp_launch_batch。
- python/sglang/srt/managers/scheduler.py (模块 调度器; 类别 source; 类型 core-logic; 符号 launch_batch_sample_if_needed, init_running_status, flush_cache, event_loop_normal): 字段定义和初始化的更改点, 以及

launch_batch_sample_if_needed 的参数化。

- python/sglang/srt/managers/scheduler_components/invariant_checker.py (模块 调度器; 类别 source; 类型 core-logic; 符号 create_scheduler_watchdog) : 看门狗创建函数中使用 cur_batch_for_debug, 确保唯一使用者正确。
- python/sglang/srt/hardware_backend/mlx/scheduler_mixin.py (模块 调度器; 类别 source; 类型 core-logic; 符号 event_loop_overlap_mlx) : MLX 循环中同步字段重命名。
- python/sglang/srt/disaggregation/decode.py (模块 调度器; 类别 source; 类型 core-logic; 符号 event_loop_normal_disagg_decode, event_loop_overlap_disagg_decode) : Disaggregation decode 事件循环中同步重命名和参数化。
- python/sglang/srt/disaggregation/prefill.py (模块 调度器; 类别 source; 类型 core-logic; 符号 event_loop_normal_disagg_prefill, event_loop_overlap_disagg_prefill) : Disaggregation prefill 事件循环中同步重命名和参数化。
- test/registered/unit/managers/test_scheduler_pause_generation.py (模块 测试; 类别 test; 类型 test-coverage) : 测试适配: cur_batch 改为 cur_batch_for_debug。
- test/registered/chunked_prefill/test_scripted_core_4gpu.py (模块 测试; 类别 test; 类型 test-coverage) : 测试适配: cur_batch 改为 cur_batch_for_debug。
- test/registered/unit/disaggregation/test_decode_queue_cleanup.py (模块 测试; 类别 test; 类型 test-coverage) : 测试适配: cur_batch 改为 cur_batch_for_debug。
- test/registered/unit/hardware_backend/mlx/test_attention_patching.py (模块 测试; 类别 test; 类型 test-coverage) : 测试适配: cur_batch 改为 cur_batch_for_debug。

关键符号: launch_batch_sample_if_needed, _pp_launch_batch

关键源码片段

python/sglang/srt/managers/scheduler_pp_mixin.py

PP 主循环的核心改造: 引入局部 cur_batch, 镜像到 cur_batch_for_debug, 并传参给 _pp_launch_batch。

```
def event_loop_pp(self: Scheduler):
    """PP 主循环, 使用局部 cur_batch 而非 self.cur_batch。"""
    self.init_pp_loop_state()
    while True:
        server_is_idle = True
        for mb_id in range(self.pp_loop_size):
            self.running_batch = self.running_mbs[mb_id]
            self.last_batch = self.last_mbs[mb_id]
            # 获取下一个 batch
            self.mbs[mb_id] = self.get_next_batch_to_run()
            self.running_mbs[mb_id] = self.running_batch
            cur_batch: Optional[ScheduleBatch] = self.mbs[mb_id]
            # 仅看门狗使用 cur_batch_for_debug
            self.cur_batch_for_debug = cur_batch
            if cur_batch:
```

```
server_is_idle = False
pp_proxy_tensors = self._pp_recv_proxy_tensors()
# ... 其他处理 ...
if cur_batch:
    # 显式传入局部变量，不依赖字段
    result, self.launch_event = self._pp_launch_batch(
        mb_id, cur_batch, pp_proxy_tensors,
        self.mb_metadata, self.last_rank_comm_queue,
    )
# ... 继续 ...
```

评论区精华

仅 `gemini-code-assist[bot]` 的一条评论指出 `launch_batch_sample_if_needed` 中新增的 `cur_batch` 参数可能为 `None`，建议类型标注为 `Optional[ScheduleBatch]` 并添加早期返回保护。该评论未获作者回应或修改，PR 已合并。

- `launch_batch_sample_if_needed` 参数可能为 `None` (correctness): 作者未回应或修改，PR 已合并，认为实际调用路径不会传入 `None`。

风险与影响

- 风险：风险较低：纯重命名 + 参数化，无行为变更。但若有外部代码直接引用 `self.cur_batch` 会引发 `AttributeError`。通过等价性审计和测试覆盖确保无遗漏。
- 影响：对用户无功能影响；对开发者，字段重命名需适应，但提升了调度器内部状态管理清晰度。团队需更新依赖 `self.cur_batch` 的私有插件或分支。
- 风险标记：字段重命名可能遗漏引用，部分调用可能传入 `None`

关联脉络

- PR #29408 Avoid implicit field-based side channel in Scheduler planning: 同作者在同一功能线上，同为消除调度器中字段隐式副作用，当前 PR 是前置步骤。