PR #25619 完整报告

sgl-project/sglang

Add SchedulerPoolStatsObserver and route pool-stats state through it

合并时间：2026-05-18 18:36

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25619

执行摘要

- 一句话：引入 SchedulerPoolStatsObserver 并调整 pool-stats 方法调用
- 推荐动作：此 PR 展示了通过数据类加静态方法解耦大型 Mixin 的模式，值得调度器相关开发者精读。注意关注后续 #25621 的实际方法迁移，确保理解完整的设计意图。

功能与动机

PR 描述指出这是 "Inplace prep for the introduce-pool-stats-observer mech move."，目标是拆分 SchedulerRuntimeCheckerMixin 中与池统计相关的逻辑到独立组件，提升可维护性和可测试性。

实现拆解

1. 定义数据类：在 pool_stats_observer.py 中新增 SchedulerPoolStatsObserver 数据类（kw_only, slots, frozen），声明所有必要依赖：tree_cache、allocator、session_controller 等，并通过 get_last_batch/get_running_batch callable 提供对可变 batch 的访问。
2. 转换方法为静态：在 scheduler_runtime_checker_mixin.py 中，将全部 pool-stats 方法（streaming_session_count, active_pool_idxs, session_held_tokens, get_pool_stats 及其内部的 _get_*_token_info 辅助）改为 @staticmethod，并将第一个参数的类型注解改为 SchedulerPoolStatsObserver。原本 self.last_batch/self.running_batch 改为通过 self.get_last_batch()/self.get_running_batch() 获取；原本私有方法（下划线开头）全部公开化。由于方法仍留在 Mixin 中，static 方法内调用其他 static 方法时使用 SchedulerRuntimeCheckerMixin.method_name(self) 的限定形式。
3. 在 Scheduler 中创建 Observer 实例：在 Scheduler.__init__ 中（is_initializing=False 之前）实例化 SchedulerPoolStatsObserver，传入全部依赖和 lambda getter。
4. 更新调用方：
   - 在 get_new_batch_prefill 和 on_idle 中，将 self.get_pool_stats() 改为 self.get_pool_stats(self.pool_stats_observer)。
   - 在 scheduler_metrics_mixin.py 的 report_prefill_stats、report_decode_stats 和 get_loads 中，类似地改为传入 observer，同时将 self._streaming_session_count() 等调用改为 self.streaming_session_count(self.pool_stats_observer)。

关键文件：

- python/sglang/srt/managers/scheduler_runtime_checker_mixin.py（模块 运行检查；类别 source；类型 core-logic；符号 streaming_session_count, active_pool_idxs, session_held_tokens, session_held_full_tokens）：核心变更文件，将 pool-stats 方法全部改为静态方法，self 类型改为 SchedulerPoolStatsObserver，使用 getter 替代直接 batch 引用，并公开私有方法。
- python/sglang/srt/managers/scheduler_components/pool_stats_observer.py（模块 池统计；类别 source；类型 core-logic；符号 SchedulerPoolStatsObserver）：新增 SchedulerPoolStatsObserver 数据类，定义 pool-stats 所需的所有依赖和 callable 接口。
- python/sglang/srt/managers/scheduler.py（模块 核心调度；类别 source；类型 dependency-wiring）：在 Scheduler.__init__中实例化 SchedulerPoolStatsObserver 并注入所有依赖，更新 get_new_batch_prefill 和 on_idle 两处调用。
- python/sglang/srt/observability/scheduler_metrics_mixin.py（模块 指标报告；类别 source；类型 core-logic）：更新指标报告方法，将 pool-stats 相关调用改为传递 pool_stats_observer。

关键符号：streaming_session_count, active_pool_idxs, get_pool_stats, report_prefill_stats, report_decode_stats, get_loads

关键源码片段

python/sglang/srt/managers/scheduler_runtime_checker_mixin.py

核心变更文件，将 pool-stats 方法全部改为静态方法，self 类型改为 SchedulerPoolStatsObserver，使用 getter 替代直接 batch 引用，并公开私有方法。

```
# 文件：python/sglang/srt/managers/scheduler_runtime_checker_mixin.py
# 展示典型的静态方法模式

@staticmethod
def active_pool_idxs(self: 'SchedulerPoolStatsObserver') -> set:
    '''计算当前 batch 占用的池索引，用于 session_held 计算。'''
    idxs = set()
    # 通过 getter 访问 last_batch / running_batch，不再直接依赖 Scheduler 实例
    for batch in [self.get_last_batch(), self.get_running_batch()]:
        if batch is None or batch.is_empty():
            continue
        for req in batch.reqs:
            if req.req_pool_idx is not None:
                idxs.add(req.req_pool_idx)
    return idxs
```

python/sglang/srt/managers/scheduler_components/pool_stats_observer.py

新增 SchedulerPoolStatsObserver 数据类，定义 pool-stats 所需的所有依赖和 callable 接口。

```
# 文件：python/sglang/srt/managers/scheduler_components/pool_stats_observer.py
# 新增的 SchedulerPoolStatsObserver 数据类，作为 pool-stats 计算的唯一依赖容器

@dataclass(kw_only=True, slots=True, frozen=True)
```

```
class SchedulerPoolStatsObserver:
    # 外部依赖（均为不可变配置或只读服务）
    tree_cache: 'BasePrefixCache'
    token_to_kv_pool_allocator: 'BaseTokenToKVPoolAllocator'
    req_to_token_pool: 'ReqToTokenPool'
    session_controller: Any # 未来可改为 SessionController 类型
    hisparse_coordinator: Any # 未来可改为 HiSparseCoordinator 类型
    is_hybrid_swa: bool
    is_hybrid_ssm: bool
    enable_hisparse: bool
    full_tokens_per_layer: Any
    swa_tokens_per_layer: Any
    max_total_num_tokens: int
    # 通过 callable 间接访问 scheduler 上易变的 batch，避免直接持有 mutable 引用
    get_last_batch: Callable
    get_running_batch: Callable
    # 注意：业务方法仍在 SchedulerRuntimeCheckerMixin 中以 @staticmethod 形式存在，
    # 未来会直接移入此类（参见后续 PR #25621）
```

评论区精华

gemini-code-assist[bot] 建议：

- 在 SchedulerPoolStatsObserver 中使用具体类型（如 SessionController、HiSparseCoordinator）替换 Any 和宽泛的 Callable。
- 为 active_pool_idxs 添加返回类型 set[int]。
- 为 _get_mamba_token_info 添加返回类型 PoolStats。 这些建议未被采纳，但若执行可提升代码的静态分析质量。
- 类型注解精确性改进 (design): 建议未被采纳，但可留待后续改进以增强静态分析。

风险与影响

- 风险：
  - 方法签名变更：所有 pool-stats 方法从实例方法改为静态，若外部未更新调用点将导致 TypeError，但仓库内调用点均已同步更新。
  - 静态方法限定引用：在静态方法中使用 SchedulerRuntimeCheckerMixin.xxx(self) 限定调用，后续将方法实际移至 Observer 时需要移除前缀，可能遗漏。
  - callable getter 依赖：通过 lambda 捕获 self.last_batch，可能引入轻微循环引用或不必要的闭包开销，但影响很小。
  - 类型注解不精确：SchedulerPoolStatsObserver 中部分字段使用 Any 类型，削弱了 IDE 和类型检查器的支持。
- 影响：
  - 用户：无影响，无功能变更。
  - 系统：池统计状态和逻辑开始从 Mixin 中解耦，为后续组件迁移奠定基础。

- 团队：开发者需要理解新的 Observer 模式和静态方法约定，后续 PR 将进一步将方法移至 Observer 中。
- 风险标记：核心路径变更，缺少测试覆盖，类型注解不足

关联脉络

- PR #25621 Move pool-stats sampling to SchedulerPoolStatsObserver: 本 PR 是前置准备，后续将 pool-stats 方法实际移动到 SchedulerPoolStatsObserver 组件的下一步 PR。