PR #25627 完整报告

sgl-project/sglang

Carve out SchedulerLoadInquirer for queue-load state

合并时间：2026-05-18 18:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25627

执行摘要

- 一句话：创建 SchedulerLoadInquirer 组件并重构队列负载查询方法
- 推荐动作：值得精读，展示了在大型类中安全提取组件的典型做法：先创建轻量容器类（dataclass + callable injection），原地修改方法签名，保证可逆性。可与同一重构链的 PR #25628、#25629 等结合理解完整迁移过程。

功能与动机

此 PR 是调度器组件化重构（Refactor chain ID: introduce-load-inquirer-prep）的一部分，旨在为将队列负载查询逻辑从庞大的 SchedulerMetricsMixin 中完整迁移到独立组件做准备，减少 Mixin 的职责，提高代码可维护性。PR body 明确说明这是 'Inplace prep for the introduce-load-inquirer mech move'。

实现拆解

1. 创建 SchedulerLoadInquirer 数据类（scheduler_components/load_inquirer.py）：定义 @dataclass，包含所有必要的依赖字段以及一系列 Callable getter（如 get_waiting_queue、get_chunked_req），用于间接访问调度器的运行时可变状态。
2. 在 Scheduler.__init__ 中实例化（scheduler.py）：在初始化 pool_stats_observer 之后创建 self.load_inquirer，通过 lambda 捕获当前调度器的属性引用，注入所有 getter。
3. 重构 SchedulerMetricsMixin 中的两个方法（scheduler_metrics_mixin.py）：将 get_loads 和 _get_num_pending_tokens 改为 @staticmethod，第一个参数类型从 Scheduler 改为 SchedulerLoadInquirer，所有对 self.waiting_queue、self.chunked_req 等直接属性访问替换为对传入 self.get_xxx() getter 的调用。
4. 更新所有调用点：在 scheduler.py 的 RPC 分派元组（init_request_dispatcher）和 _get_new_batch_prefill_raw 中通过 self.load_inquirer 传递新参数；在 scheduler_output_processor_mixin.py 的 stream_output_generation 中同样调整调用方式。方法体物理移动留待后续 introduce-load-inquirer-move 提交。

关键文件：

- python/sglang/srt/observability/scheduler_metrics_mixin.py（模块 观察层；类别 source；类型 core-logic；符号 _get_num_pending_tokens, get_loads）：核心修改文件：_get_num_pending_tokens 和 get_loads 从实例方法改为静态方法，签名从 Scheduler 改为 SchedulerLoadInquirer，内部属性访问替换为 getter 调用。

- python/sglang/srt/managers/scheduler_components/load_inquirer.py（模块 调度器；类别 source；类型 core-logic；符号 SchedulerLoadInquirer）：新文件：定义 SchedulerLoadInquirer 数据类，集中声明所有依赖和 Callable getter 字段，是后续迁移的目标容器。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 dependency-wiring）：注入 SchedulerLoadInquirer 实例，并更新两处调用点（RPC 分派和 _get_new_batch_prefill_raw）。
- python/sglang/srt/managers/scheduler_output_processor_mixin.py（模块 输出处理；类别 source；类型 core-logic）：调用 get_loads 时增加 self.load_inquirer 参数。

关键符号：SchedulerLoadInquirer, _get_num_pending_tokens, get_loads

关键源码片段

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

新文件：定义 SchedulerLoadInquirer 数据类，集中声明所有依赖和 Callable getter 字段，是后续迁移的目标容器。

```
@dataclass(kw_only=True, slots=True, frozen=True)
class SchedulerLoadInquirer:
    """调度器负载查询器，封装队列负载相关的状态和策略。"""

    # 依赖注入：使用 Callable 获取运行时可变状态，避免直接持有引用
    disaggregation_mode: "DisaggregationMode"
    ps: "ParallelState"
    server_args: "ServerArgs"
    max_total_num_tokens: int
    max_running_requests: int
    pool_stats_observer: "SchedulerPoolStatsObserver"
    tp_worker: "BaseTpWorker"
    token_to_kv_pool_allocator: "BaseTokenToKVPoolAllocator"
    spec_algorithm: "SpeculativeAlgorithm"

    # 以下为 Callable getter，用于获取 Scheduler 中的运行时状态
    get_running_batch: Callable
    get_waiting_queue: Callable
    get_stats: Callable
    get_chunked_req: Callable
    get_disagg_prefill_bootstrap_queue: Callable
    get_disagg_prefill_inflight_queue: Callable
    get_disagg_decode_prealloc_queue: Callable
    get_disagg_decode_transfer_queue: Callable
    get_spec_total_num_accept_tokens: Callable
    get_spec_total_num_forward_ct: Callable
```

评论区精华

未发现 review 评论记录。作者在 PR body 中明确解释了设计决策：采用 Callable injection 而非 per-call kwarg 是为了保持 get_loads/_get_num_pending_tokens 签名稳定，使后续 'move' 提交成为真正的字节级剪切粘贴，该模式与 SchedulerMetricsReporter 一致。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 签名变更遗漏：方法签名从 Scheduler 改为 SchedulerLoadInquirer，若存在未扫描到的调用点（如第三方补丁）可能引发类型错误。
  2. 间接层性能：getter 均为 lambda 闭包，每次调用增加一次函数调用开销，但负载查询不在热路径中，影响可忽略。
  3. 无测试覆盖：本次未新增或修改测试，后续迁移步骤若行为稍有偏差可能无法被 CI 捕获。 - 影响：对用户和外部 API 无影响，行为完全一致。对开发团队，代码结构更清晰，SchedulerMetricsMixin 职责减轻，为后续拆解 Mixin 奠定基础。对基础设施无影响。 - 风险标记：核心路径变更，无测试覆盖，Callable injection 引入间接层

关联脉络

- PR #25628 Move queue-load reporting to SchedulerLoadInquirer: 此 PR 的后续步骤，将方法体物理迁移到 SchedulerLoadInquirer，依赖本 PR 的骨架和调用点准备。
- PR #25629 Add SchedulerMetricsReporter and route metrics state through it: 同一重构链中的类似提取模式，使用相同的 Callable injection 手法，可对比学习。