PR #25441 完整报告

sgl-project/sglang

Annotate dead max_running_requests_under_SLO

合并时间：2026-05-16 09:22

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25441

执行摘要

- 一句话：标记死代码字段并添加 TODO 注释
- 推荐动作：值得快速合并，因为它提前标记了指标失效的根因，便于后续有人修复合入时应置一个 setter。建议后续 PR 修复 regression（在 __init__ 或对应配置更新处添加赋值），并考虑为 sglang:utilization 补充单元测试。

功能与动机

PR body 指出 max_running_requests_under_SLO 在 regression (#22713) 后失去了 setter，导致两个读取点始终回退到 None，进而使 sglang:utilization 指标卡在 0。当前提交仅添加注释，不修改行为，以避免在一次 commit 中混入机械标记和功能性修复。

实现拆解

1. python/sglang/srt/managers/scheduler.py：在 init_model_worker 方法中 emit_constants 调用处，为 max_running_requests_under_SLO 参数添加了 # TODO 注释，说明该属性没有 setter，属于死链。
2. python/sglang/srt/observability/scheduler_metrics_mixin.py：在 calculate_utilization 方法的 getattr 调用前添加了更详细的 # TODO 注释，指出这一死链导致 sglang:utilization 指标永远为 0，并注明 regression 来源为 #22713。
3. 两个改动各行只有一行注释新增，+1/-0，完全不涉及逻辑变更，属于纯机械的标记性提交。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：标记了 metrics collector 中 emit_constants 调用的死属性参数
- python/sglang/srt/observability/scheduler_metrics_mixin.py（模块 可观测性；类别 source；类型 core-logic）：标记了 utilization 计算中读入的死属性，并指明 sglang:utilization 卡 0 的根因

关键符号：init_model_worker, calculate_utilization

关键源码片段

python/sglang/srt/managers/scheduler.py

标记了 metrics collector 中 emit_constants 调用的死属性参数

```
def init_model_worker(self):
    # ... 省略其他初始化代码 ...
    if self.enable_metrics:
        self.metrics_collector.emit_constants(
            max_total_num_tokens=self.max_total_num_tokens,
            # TODO: max_running_requests_under_SLO 没有 setter——死链。
            max_running_requests_under_SLO=getattr(
                self, "max_running_requests_under_SLO", None
            ),
            engine_startup_time=0.0,
            engine_load_weights_time=0.0,
            page_size=self.page_size,
            num_pages=self.max_total_num_tokens // self.page_size,
            context_len=self.model_config.context_len,
            startup_available_gpu_memory_gb=avail_mem,
        )
```

python/sglang/srt/observability/scheduler_metrics_mixin.py

标记了 utilization 计算中读入的死属性，并指明 sglang:utilization 卡 0 的根因

```
def calculate_utilization(self: Scheduler):
    if self.disaggregation_mode == DisaggregationMode.PREFILL:
        self.stats.utilization = -1
    else:
        # TODO: max_running_requests_under_SLO 没有 setter——
        # sglang:utilization 卡在 0（由 #22713 引入）。
        max_under_slo = getattr(self, "max_running_requests_under_SLO", None)
        if max_under_slo is not None and max_under_slo > 0:
            self.stats.utilization = max(
                self.stats.num_running_reqs.total / max_under_slo,
                self.stats.token_usage / 0.9,
            )
```

评论区精华

该 PR 没有 reviewer 评论，只有一个来自 gemini-code-assist[bot] 的 quota 超限提示，无实质讨论。

- 暂无高价值评论线程

风险与影响

- 风险：本 PR 只添加注释，无任何运行时行为变更，风险极低。真正的风险在于后续修复可能被遗忘。
- 影响：对用户无直接影响，对维护者有正面影响：通过注释清晰标记了死代码的坐标和根因，降低了未来维护和修复的心智负担。
- 风险标记：暂无

关联脉络

- PR #22713 未知：PR body 指出 #22713 引入了 regression，导致 max_running_requests_under_SLO 失去了 setter。本 PR 的注释正是为后续修复该 regression 做准备。