

PR #25622 完整报告

sgl-project/sglang

Move create_scheduler_watchdog from runtime_checker mixin to scheduler.py

合并时间: 2026-05-18 18:38

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25622>

执行摘要

- 一句话: 将 create_scheduler_watchdog 迁移到 scheduler.py
- 推荐动作: 值得精读, 特别是关注 MECH_COMMIT_SPLIT.md 中定义的拆分策略。该 PR 展示了如何安全地进行纯机械代码搬迁, 但应关注 review 中提出的竞态问题是否需要后续 PR 中修复。

功能与动机

为后续引入 SchedulerInvariantChecker 组件的机械拆分做预准备, 将模块级自由函数从 mixin 文件迁出, 以保持文件职责清晰。

实现拆解

1. 在 scheduler.py 的导入块中, 将 create_scheduler_watchdog 从 scheduler_runtime_checker_mixin 的导入列表中移除, 并新增 from sglang.srt.utils.watchdog import WatchdogRaw。
2. 在 scheduler.py 中 validate_dflash_request 函数之后、class Scheduler 之前, 原样粘贴 create_scheduler_watchdog 函数定义, 并将参数类型注解 scheduler: Scheduler 改为字符串前向引用 scheduler: "Scheduler", 以避免 Python 运行时名称错误。
3. 在 scheduler_runtime_checker_mixin.py 中删除整个 create_scheduler_watchdog 函数定义及其内部闭包 dump_info; 同时删除文件中不再使用的 from sglang.srt.utils.watchdog import WatchdogRaw 导入。

关键文件:

- python/sglang/srt/managers/scheduler.py (模块 调度器; 类别 source; 类型 core-logic; 符号 create_scheduler_watchdog, dump_info) : 接收函数代码并调整导入, 是主要变更文件。
- python/sglang/srt/managers/scheduler_runtime_checker_mixin.py (模块 调度器; 类别 source; 类型 core-logic; 符号 create_scheduler_watchdog, dump_info) : 移除了 create_scheduler_watchdog 函数和相关导入, 清理了代码。

关键符号: create_scheduler_watchdog, dump_info

关键源码片段

python/sglang/srt/managers/scheduler.py

接收函数代码并调整导入，是主要变更文件。

```
# python/sglang/srt/managers/scheduler.py

# 导入部分：新增 WatchdogRaw，移除 create_scheduler_watchdog
from sglang.srt.utils.watchdog import WatchdogRaw

# ... 中间代码保持不变 ...

# 在 class Scheduler 定义之前，粘贴 create_scheduler_watchdog 函数
# 注意：使用字符串前向引用 "Scheduler" 以规避 Python 编译时名称错误
def create_scheduler_watchdog(
    scheduler: "Scheduler", watchdog_timeout: float, soft: bool = False
) -> WatchdogRaw:
    """为调度器创建一个看门狗实例，用于检测主循环是否卡住。"""
    def dump_info() -> str:
        # 如果调度器还在初始化，则返回空字符串
        if scheduler.is_initializing:
            return ""
        # 获取所有内存池的检查信息
        _, messages = scheduler._check_all_pools(
            scheduler.pool_stats_observer.get_pool_stats()
        )
        return (
            f"{scheduler.cur_batch.batch_size()=}\n"
            f"{scheduler.cur_batch.reqs=}\n" + "\n".join(messages)
        )

    return WatchdogRaw(
        debug_name="Scheduler",
        get_counter=lambda: scheduler.forward_ct,
        is_active=lambda: scheduler.is_initializing or scheduler.cur_batch is not None,
        watchdog_timeout=watchdog_timeout,
        soft=soft,
        dump_info=dump_info,
    )
```

python/sglang/srt/managers/scheduler_runtime_checker_mixin.py

移除了 create_scheduler_watchdog 函数和相关导入，清理了代码。

```
# python/sglang/srt/managers/scheduler_runtime_checker_mixin.py

# 移除的导入行：
# from sglang.srt.utils.watchdog import WatchdogRaw

# 移除的函数定义（约 25 行）：
# def create_scheduler_watchdog(...):
# def dump_info() -> str:
```

```
# ...  
# return WatchdogRaw(...)
```

评论区精华

Review 中 `gemini-code-assist[bot]` 指出 `dump_info` 函数中 `scheduler.cur_batch` 可能为 `None`（在后台看门狗线程中访问时存在竞态条件），建议添加防御性判空检查。该评论结论未被采纳或解决（PR 已合并）。

- `dump_info` 中 `cur_batch` 的竞态条件 (correctness): 建议添加防御性判空检查，但 PR 已合并，该问题尚未修复。

风险与影响

- 风险：主要风险在于 `dump_info` 中访问 `scheduler.cur_batch` 可能因竞态条件而触发 `AttributeError`，导致看门狗线程崩溃，进而影响调度器的故障检测机制。该问题在 review 中被指出但未在 PR 中修复。
- 影响：对用户无直接影响，属于纯内部重构。对系统的影响仅限于调度器启动时看门狗的创建位置变更，功能不变。对团队后续可维护性有正向作用，为进一步拆分 `SchedulerInvariantChecker` 组件扫平障碍。
- 风险标记：线程竞态未修复

关联脉络

- PR #25623 Introduce `SchedulerInvariantChecker` to own invariant-check state: 该 PR 是此 PR 的直接后续，引入 `SchedulerInvariantChecker` 组件，本 PR 是为其做前置搬迁。
- PR #25624 Move invariant checks to `SchedulerInvariantChecker` and retire `runtime_checker` mixin: 进一步完善拆分，将不变量检查逻辑全部从 `mixin` 迁出。