PR #25716 完整报告

sgl-project/sglang

Pack scattered new-token-ratio state into a dedicated tracker

合并时间：2026-05-19 09:19

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25716

执行摘要

- 一句话：将分散的 new_token_ratio 状态封装为专用类
- 推荐动作：此 PR 是一个干净的重构，值得合并。建议后续为此类添加单元测试，确保状态转换的正确性。

功能与动机

Scheduler 原本使用四个分散的属性来管理 new_token_ratio 的状态，在多个调用点散布开放编码的算术逻辑（如 max(... - decay, min)）。通过封装为专用追踪器类，将状态转换转为命名方法，减少重复代码，降低引入逻辑错误的风险。

实现拆解

1. 新增 NewTokenRatioTracker 类（new_token_ratio_tracker.py）：使用 dataclass(slots=True, kw_only=True) 定义四个字段（init, min, decay, current），提供 from_server_args 类方法根据环境变量和 ServerArgs 计算初始值，实现 decay_step() 和 reset() 方法封装状态转换。
2. 修改 Scheduler 初始化（scheduler.py）：将原 4 个属性替换为一个 self.new_token_ratio_tracker 属性，调用 NewTokenRatioTracker.from_server_args(self.server_args) 完成初始化。
3. 修改所有调用点（scheduler.py 和 dllm/mixin/scheduler.py）：将原 self.new_token_ratio 的使用替换为 self.new_token_ratio_tracker.current；将 decay 逻辑替换为 self.new_token_ratio_tracker.decay_step()；将 reset 逻辑替换为 self.new_token_ratio_tracker.reset()。
4. 无测试配套变更：本次重构未添加或修改测试文件，但原有行为完全保持。

关键文件：

- python/sglang/srt/managers/scheduler_components/new_token_ratio_tracker.py（模块 调度器；类别 source；类型 core-logic；符号 NewTokenRatioTracker, from_server_args, decay_step, reset）：核心新增文件，定义了 NewTokenRatioTracker 类及其工厂方法和状态转换方法。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 dependency-wiring）：Scheduler 主文件中移除旧的四个属性，替换为 NewTokenRatioTracker 实例，并更新所有调用点。

- python/sglang/srt/dllm/mixin/scheduler.py（模块 调度器；类别 source；类型 core-logic）：DLLM 调度 mixin 中一处 PrefillAdder 创建时引用 new_token_ratio 改为 new_token_ratio_tracker.current。

关键符号：NewTokenRatioTracker.from_server_args, NewTokenRatioTracker.decay_step, NewTokenRatioTracker.reset

关键源码片段

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

核心新增文件，定义了 NewTokenRatioTracker 类及其工厂方法和状态转换方法。

```
from dataclasses import dataclass

from sglang.srt.environ import envs
from sglang.srt.server_args import ServerArgs


@dataclass(slots=True, kw_only=True)
class NewTokenRatioTracker:
    # 用于管理 KV 预算头 room 因子的状态
    init: float # 初始比率，由环境变量和 schedule_conservativeness 决定
    min: float # 最小比率，避免过度回退
    decay: float # 每次非 retract 步骤的衰减步长
    current: float # 当前活跃比率

    @classmethod
    def from_server_args(cls, server_args: ServerArgs) -> "NewTokenRatioTracker":
        # 根据 ServerArgs 和环境变量计算初始值
        init = min(
            envs.SGLANG_INIT_NEW_TOKEN_RATIO.get()
            * server_args.schedule_conservativeness,
            1.0,
        )
        min_ratio = min(
            init * envs.SGLANG_MIN_NEW_TOKEN_RATIO_FACTOR.get(),
            1.0,
        )
        decay = (init - min_ratio) / envs.SGLANG_NEW_TOKEN_RATIO_DECAY_STEPS.get()
        return cls(init=init, min=min_ratio, decay=decay, current=init)

    def decay_step(self) -> None:
        # 每次调度步骤后向最小值衰减
        self.current = max(self.current - self.decay, self.min)

    def reset(self) -> None:
        # 调度器空闲时重置为初始值
        self.current = self.init
```

评论区精华

该 PR 仅有 1 条评论，来自 gemini-code-assist[bot] 的配额提醒，无实际技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 回归风险：所有调用点均已更新，但若未来其他模块直接访问已删除的属性（如 self.new_token_ratio），将导致 AttributeError。由于这些属性是 Scheduler 的内部属性，影响面可控。
  2. 类型安全：NewTokenRatioTracker 使用 slots=True，禁止动态添加属性，运行时若不慎赋值会报错，但属于预期行为。
  3. 无性能影响：封装未引入额外开销，dataclass 与普通类性能相当。
- 影响：
  1. 代码可维护性提升：new_token_ratio 相关的状态转换从散落的开放代码变为集中的命名方法，降低认知负荷和 bug 概率。
  2. 影响范围：仅修改 scheduler.py、scheduler_components 新增文件、dllm mixin 中的一处调用，副作用极小。
  3. 对用户透明：无功能变更，调度行为完全一致。 - 风险标记：缺少测试覆盖

关联脉络

- PR #25717 Move the retract-decode ratio estimation onto the new-token-ratio tracker: 后续 PR，在此重构基础上进一步将 retract-decode 比率估计迁移到追踪器中。