PR #22247 完整报告

sgl-project/sglang

[Anthropic] Fix clock mismatch in received_time causing negative Prometheus metrics

合并时间：2026-04-14 12:22

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22247

执行摘要

本 PR 修复了 Anthropic API 入口因使用挂钟时间（time.time()）而非单调时间（monotonic_time()）记录请求接收时间，导致下游 Prometheus 指标（如 TTFT 和端到端延迟）出现大负值的问题。变更仅影响指标收集，不影响模型输出，是 OpenAI 入口类似修复的扩展，提升生产环境监控准确性。

功能与动机

问题背景：Issue #22249 报告 Anthropic API 入口（serving.py）使用 time.time()（挂钟时间，约 1.7e9）记录 received_time，而下游 req_time_stats.py 使用 time.perf_counter()（单调时间，约 1e5）计算延迟差值，时钟不匹配导致 Prometheus 直方图出现大负值。生产环境中发现 Anthropic 请求在 Grafana 仪表板持续显示负 TTFT 和延迟。

关联修复：OpenAI 入口已在 PR #17640 修复为使用 monotonic_time()，但 Anthropic 入口被遗漏。本 PR 旨在统一时钟使用，确保指标准确性。

实现拆解

修改文件：python/sglang/srt/entrypoints/anthropic/serving.py

变更点	修改前	修改后	说明
导入	-	from sglang.srt.observability.req_time_stats import monotonic_time	引入单调时间函数
非流处理（第 317 行）	received_time = time.time()	received_time = monotonic_time()	使用单调时钟记录接收时间
流处理（第 373 行）	received_time = time.time()	received_time = monotonic_time()	同上

关键细节：

- received_time_perf 变量保持不变，用于本地 validation_time 计算。

- monotonic_time() 是 time.perf_counter() 的别名，确保与下游指标计算时钟一致。

评论区精华

gemini-code-assist[bot] 在 review 中提出优化建议：

> 由于 monotonic_time 是 time.perf_counter() 的别名，received_time 和 received_time_perf 现在持有相同值。应考虑合并赋值以避免冗余时钟调用并提高清晰度，例如：received_time = received_time_perf = monotonic_time()。

但 PR 作者未采纳该建议，最终代码保留两个独立变量。可能原因：received_time_perf 用于本地验证时间计算，但未在讨论中明确说明。JustinTong0323 直接批准 PR。

风险与影响

风险分析：

- 极低风险：仅更改时间戳来源，不影响核心请求处理或模型输出。
- 使用单调时钟避免挂钟调整（如 NTP 同步）影响延迟测量，提升指标准确性。
- 潜在轻微性能开销：未合并冗余赋值导致两次调用同一函数，但影响可忽略。

影响评估：

- 用户：修复生产监控指标，使 Anthropic API 的 TTFT 和端到端延迟指标恢复为正确定值。
- 系统：仅影响 Prometheus 指标收集，不改变请求处理行为。
- 团队：统一 Anthropic 和 OpenAI 入口的时钟使用，减少维护不一致性。

关联脉络

- PR #17640：OpenAI 入口的相同修复，本 PR 扩展至 Anthropic 入口，体现跨入口一致性维护。
- PR #22726：同属 observability 标签，涉及 Prometheus 指标改进，反映团队对可观测性的持续投入。
- PR #21646：修改相同文件 req_time_stats.py，涉及指标计算优化，显示该模块的活跃演进。

整体趋势：sglang 项目近期多个 PR 关注监控指标准确性（如 #22726、#22331、#22506），本 PR 是这一方向的延续，强调生产环境可观测性的重要性。