PR #24521 完整报告

sgl-project/sglang

Improve metrics, observability, and PD deploy tooling

合并时间：2026-05-07 02:27

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24521

执行摘要

- 一句话：增强可观测性：细化队列时间桶、新增指标、修复吞吐计算
- 推荐动作：建议对可观测性系统负责的工程师精读此 PR。特别关注 decode_throughput 修复背景和 uncached prompt tokens 指标设计。对于运维团队，建议验证日志解析器是否兼容新格式。整体变更价值高，值得引入。

功能与动机

为了更精确地监控请求队列延迟和未缓存 token 长度，修复错误的解码吞吐计算，并统一可观测性基础设施。PR body 明确指出需要 finer-grained queueing time histogram buckets（sub-100ms granularity）和 uncached prompt tokens histogram metric。

实现拆解

1. 重构请求时间统计 (req_time_stats.py)：移除未使用的 prefill_run_batch_start_time / end_time 字段及相关 set_time_batch 调用；修复 decode_throughput 计算，将分子从 completion_tokens 改为 completion_tokens - 1 以反映实际解码 token 数；将 spec 追踪调用包装在 tracing_enable 标志内以减少非追踪模式下的开销；移除 inference_time 元数据字段（原依赖 forward_entry_time）。
2. 增强指标采集器 (metrics_collector.py)：为 queue_time_seconds 直方图添加亚 100ms 粒度桶（起始值 0.000, 0.001, 0.005, 0.010, 0.050, 0.100, 0.200, 0.500），提升短延迟测量精度；新增 uncached_prompt_tokens_histogram 指标，上报 prompt_tokens - cached_tokens，便于分析计算量。
3. 改进日志格式：在 schedule_batch.py 的 log_time_stats 方法中将日志前缀从 "Req Time Stats" 改为 "ReqTimeStats"，并新增 cached_input_len 字段；在 log_utils.py 中简化 stdout 日志格式，移除时间戳前缀（由框架自动添加）；在 request_logger.py 的跳过列表中添加 video_data。
4. 字段顺序调整与清理：在 io_struct.py 中将 positional_embed_overrides 移动到 custom_logit_processor 之后，保持字段顺序一致性；在 tokenizer_manager.py 中重置 fwd_occupancy 计数器。
5. 测试与部署工具：更新 test_log_utils.py 中的测试用例，支持解析新的日志前缀；在 test_disaggregation_different_tp.py 中启用指标和请求时间统计记录，验证 PD 场景下的可观测性。

关键文件：

- python/sglang/srt/observability/req_time_stats.py（模块 请求时间统计；类别 source；类型 core-logic；符号 set_prefill_run_batch_start_time, set_prefill_run_batch_end_time, get_prefill_waiting_latency, get_prefill_launch_latency）：核心修改：移除 prefill_run_batch 时间跟踪，修复 decode_throughput 计算，保护 spec 追踪调用
- python/sglang/srt/observability/metrics_collector.py（模块 指标采集器；类别 source；类型 core-logic）：新增 uncached_prompt_tokens_histogram 指标，细化队列时间桶
- python/sglang/srt/managers/schedule_batch.py（模块 批次管理；类别 source；类型 core-logic）：改进 ReqTimeStats 日志前缀，新增 cached_input_len 字段
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：移除预填充批处理时间采集点，简化 run_batch
- python/sglang/srt/managers/io_struct.py（模块 请求结构；类别 source；类型 core-logic）：调整 positional_embed_overrides 字段顺序以保持一致性
- test/registered/utils/test_log_utils.py（模块 日志工具测试；类别 test；类型 test-coverage；符号 _parse_log_json）：适配新日志格式，新增 _parse_log_json 工具函数
- python/sglang/srt/utils/log_utils.py（模块 日志工具；类别 source；类型 core-logic）：简化 stdout 日志格式，移除时间戳前缀
- python/sglang/srt/observability/scheduler_metrics_mixin.py（模块 指标混入；类别 source；类型 core-logic）：新增一行配置，支持视频数据跳过
- python/sglang/srt/managers/tokenizer_manager.py（模块 Token 管理器；类别 source；类型 core-logic）：重置 fwd_occupancy 计数器
- python/sglang/srt/utils/request_logger.py（模块 请求日志；类别 source；类型 core-logic）：跳过视频数据，避免日志过大
- test/registered/distributed/test_disaggregation_different_tp.py（模块 PD 差异 TP 测试；类别 test；类型 test-coverage）：启用指标和请求时间统计记录，覆盖 PD 场景

关键符号：convert_to_output_meta_info, log_time_stats, set_prefill_run_batch_start_time, set_prefill_run_batch_end_time, _parse_log_json

关键源码片段

python/sglang/srt/observability/req_time_stats.py

核心修改：移除 prefill_run_batch 时间跟踪，修复 decode_throughput 计算，保护 spec 追踪调用

```
# 修复后的 convert_to_output_meta_info：移除了 inference_time 字段，修复 decode_throughput 计算
def convert_to_output_meta_info(
    self, scheduler_time_stats=None, completion_tokens=0
) -> dict:
    meta_info = {}
```

```
    if self.created_time > 0.0:
        meta_info["request_received_ts"] = convert_time_to_realtime(self.created_time)
    if self.api_server_dispatch_finish_time > 0.0:
        meta_info["api_server_dispatch_finish_ts"] = convert_time_to_realtime(self.api_server_
        dispatch_finish_time)
    if self.response_sent_to_client_time > 0.0:
        meta_info["response_sent_to_client_ts"] = convert_time_to_realtime(self.response_sent_to_
        client_time)
    if self.finished_time > 0.0:
        meta_info["request_finished_ts"] = convert_time_to_realtime(self.finished_time)
    # 注意：移除了 inference_time 字段，原依赖 scheduler_time_stats.forward_entry_time
    # 修复 decode_throughput：使用 (completion_tokens - 1) 以排除首个 token（prefill 输出）
    decode_latency = self.get_decode_latency()
    if decode_latency > 0.0 and completion_tokens > 1:
        meta_info["decode_throughput"] = (completion_tokens - 1) / decode_latency
    return meta_info
```

python/sglang/srt/observability/metrics_collector.py

新增 uncached_prompt_tokens_histogram 指标，细化队列时间桶

```
# 在 __init__ 中新增 uncached_prompt_tokens_histogram，并细化 queue_time 桶
# queue_time 桶添加亚 100ms 级别：0.000, 0.001, 0.005, 0.010, 0.050, 0.100, 0.200, 0.500
self.uncached_prompt_tokens_histogram = Histogram(
    name="sglang:uncached_prompt_tokens_histogram",
    documentation="Histogram of uncached (compute) prompt token length.",
    labelnames=labels.keys(),
    buckets=generate_buckets(
        server_args.prompt_tokens_buckets, default_bucket_prompt_tokens
    ),
)
# 在 log_request_stats 中上报 uncached token 长度
self.uncached_prompt_tokens_histogram.labels(**labels).observe(
    float(prompt_tokens - cached_tokens) # cached_tokens 由调用方传入
)
```

评论区精华

唯一的 review comment 来自 gemini-code-assist[bot]，指出在 metrics_collector.py 第 1508 行使用了未定义变量 cached_tokens，可能导致 NameError。该 comment 标记为高优先级，但 PR 已合并未见修复。可能 cached_tokens 由调用方传入（在 report_cache_source 方法中可见），但代码片段中未明确体现，需确认。

- 未定义变量 cached_tokens (correctness): PR 已合并但未见修复，可能 cached_tokens 由调用方传入，但代码片段中未明确体现。需进一步确认。

风险与影响

- 风险：

    - 字段移除风险：移除 prefill_run_batch_start_time/end_time 可能导致依赖这些字段的外部代码报错，但仓库内已无条件引用。
    - 计算公式更改：decode_throughput 从 completion_tokens / decode_latency 改为 (completion_tokens - 1) / decode_latency，可能影响依赖此值的监控面板或告警。
    - 日志格式变更：ReqTimeStats 前缀改变和 cached_input_len 加入可能破坏基于文本解析的日志分析管道。
    - 未定义变量风险：metrics_collector.py 中 cached_tokens 可能未定义（需确认调用方是否传递）。
    - 性能开销：队列时间桶更细粒度可能轻微增加 Prometheus 存储开销，但可接受。
- 影响：
    - 用户影响：无直接用户可见变化，但 API 返回的 meta_info 中不再包含 inference_time 字段（如果有用户依赖此字段需更新）。
    - 系统影响：增强的可观测性有助于运维团队更准确地进行性能分析和容量规划。
    - 团队影响：需更新内部监控仪表盘以利用新指标（uncached_prompt_tokens，细粒度队列时间）并调整 decode_throughput 阈值。
    - 风险标记：字段移除，公式修正，日志格式变更，未定义变量风险

关联脉络

- PR #24522 [PD] Fix missing update_status call in abort() across all KV backends: 同样涉及 PD 可观测性，确保状态同步与本 PR 的 metrics 改进相辅相成。
- PR #24416 [PD] Fix KV transfer metrics: 直接修复 KV 传输指标，与本 PR 改善 metrics 方向一致。