PR #25163 完整报告

sgl-project/sglang

Add sglang:get_loads_duration_seconds metric

合并时间：2026-05-14 09:38

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25163

执行摘要

- 一句话：为 /v1/loads 端点添加 Prometheus 延迟直方图
- 推荐动作：该 PR 变更小、影响明确，可作为可观测性增强的范例。建议精读 v1_loads.py 的 finally 块实现，学习 try/finally 埋点模式。虽然 review 建议未采纳，但整体逻辑正确，可以直接合入。

功能与动机

PR body 明确指出：「Add a Histogram metric (sglang:get_loads_duration_seconds) to track the latency of /v1/loads requests」。目的是增强对负载端点延迟的可观测性，便于负载均衡和容量规划。

实现拆解

1. 注册直方图指标：在 python/sglang/srt/observability/metrics_collector.py 的 TokenizerMetricsCollector.__init__ 中，新增一个 Histogram 实例 self.get_loads_duration_seconds，使用与已有指标相同的标签集，并设置细粒度的亚秒级 buckets（0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0）。
2. 端点埋点计时：在 python/sglang/srt/entrypoints/v1_loads.py 的 get_loads 路由中，在方法入口处记录 time.perf_counter() 起始时间，在 try/finally 块中获取 tokenizer_manager.metrics_collector，并通过 mc.get_loads_duration_seconds.labels(**mc.labels).observe(...) 上报耗时。
3. 测试配套修改：在 test/registered/language/test_srt_backend.py 的 setUpClass 中，为 sgl.Runtime 添加 enable_metrics=True 参数，使得测试服务器运行时启用指标收集，从而自动覆盖新注册的直方图。

关键文件：

- python/sglang/srt/entrypoints/v1_loads.py（模块 负载端；类别 source；类型 dependency-wiring；符号 get_loads）：核心变更点：在 /v1/loads 端点中添加计时逻辑，引入 time 模块，在 try/finally 块中记录指标。
- python/sglang/srt/observability/metrics_collector.py（模块 可观测性；类别 source；类型 core-logic；符号 TokenizerMetricsCollector.init）：注册 Histogram 指标的定义，包括指标名称、文档和 buckets。

- test/registered/language/test_srt_backend.py（模块 测试；类别 test；类型 test-coverage；符号 TestSRTBackend.setUpClass）：测试启用 metrics，确保新指标在测试环境中被覆盖。

关键符号：get_loads, TokenizerMetricsCollector.init, TestSRTBackend.setUpClass

关键源码片段

python/sglang/srt/entrypoints/v1_loads.py

核心变更点：在 /v1/loads 端点中添加计时逻辑，引入 time 模块，在 try/finally 块中记录指标。

```
# python/sglang/srt/entrypoints/v1_loads.py（关键变更）
import time # 新增：用于计时
# ...

@router.get("/v1/loads")
async def get_loads(
    dp_rank: Optional[int] = None,
    include: Optional[str] = None,
    format: Optional[str] = None,
    tokenizer_manager=Depends(_get_tokenizer_manager),
):
    include_list = [s.strip() for s in include.split(",")] if include else None

    start = time.perf_counter() # 新增：开始计时
    try:
        load_results = await tokenizer_manager.get_loads(
            include=include_list,
            dp_rank=dp_rank,
        )
    except ValueError as e:
        raise HTTPException(status_code=400, detail=str(e))
    finally:
        # 获取 metrics collector 并上报耗时，确保异常路径也能记录
        mc = getattr(tokenizer_manager, "metrics_collector", None)
        if mc is not None:
            mc.get_loads_duration_seconds.labels(**mc.labels).observe(
                time.perf_counter() - start
            )
    # ... 剩余逻辑不变
```

python/sglang/srt/observability/metrics_collector.py

注册 Histogram 指标的定义，包括指标名称、文档和 buckets。

```
# python/sglang/srt/observability/metrics_collector.py
# 在 TokenizerMetricsCollector.__init__ 中新增（位于已有指标注册区后）：
self.get_loads_duration_seconds = Histogram(
    name="sglang:get_loads_duration_seconds",
    documentation="Time spent serving /v1/loads requests (seconds).",
```

```
        labelnames=labels.keys(),
        buckets=(0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0),
        # 亚秒级桶，适合低延迟接口
    )
```

评论区精华

gemini-code-assist[bot] 在 review 中提出了两个改进建议：

- 在 v1_loads.py 的 finally 块中，除了检查 mc 非空外，还应使用 hasattr(mc, 'get_loads_duration_seconds') 防止 metric 属性不存在导致 AttributeError。
- 在 metrics_collector.py 中注册 Histogram 时应使用 self.labels.keys() 而非 labels.keys()，以兼容 labels 参数为 None 的情况。 这些建议均为中等优先级，但 PR 作者没有采纳，直接合并。
- 缺少 hasattr 检查可能导致 AttributeError (correctness): PR 作者未采纳，直接合并。项目内 metric collector 统一，风险低。
- Histogram labelnames 应使用 self.labels.keys() (correctness): PR 作者未采纳，直接合并。实际调用中 labels 不会为 None。

风险与影响

- 风险：
  1. 低风险：变更仅在 /v1/loads 端点中增加了计时逻辑和指标注册，不修改核心请求处理路径。
  2. 潜在 AttributeError：未采纳 review 建议，如果 metrics_collector 对象不存在 get_loads_duration_seconds 属性，可能抛出异常。但鉴于 metrics_collector 是该项目内统一的指标采集器，风险可控。
  3. 标签键不一致：未使用 self.labels.keys()，若 labels 参数为 None 则可能引发错误，但实际调用时 labels 均由 TokenizerMetricsCollector 内部传入，不会为 None。
- 影响：
  1. 用户：无直接影响。新增指标仅影响 Prometheus 监控端点 /metrics，用户可通过该端点观察 /v1/loads 延迟。
  2. 系统：增加微小的内存开销（注册一个 Histogram）和运行时开销（一次 time.perf_counter() 调用），可忽略不计。
  3. 团队：丰富了监控指标，有助于排查负载均衡和调度相关问题。 - 风险标记：review 建议未采纳，潜在 AttributeError

关联脉络

- PR #24858 multi_layer_eagle: add tracing hooks: 同属可观测性增强，在 spec decode 模块添加追踪钩子。