

PR #27363 完整报告

sgl-project/sglang

[srt] Add sglang:weight_load_duration_seconds gauge with source label

合并时间: 2026-06-08 23:41

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27363>

执行摘要

- 一句话: 新增权重加载耗时 Prometheus gauge 指标, 区分四类来源
- 推荐动作: 建议合并。设计清晰: 采用上下文管理器复用计时逻辑, source 标签区分来源, multiprocess_mode='mostrecent' 适配 Prometheus 多进程模式。防御性处理可选 collector, 确保向后兼容。

功能与动机

"update_weights_from_disk|distributed|tensor|ipc calls happen while the engine is paused, so the periodic log_stats path can't carry their latency. Without a dedicated metric there's no way to observe how long weight updates take per rank — useful for tail-latency dashboards and for distinguishing the four update paths in production."

实现拆解

实现步骤:

1. 在 metrics_collector.py 中定义 Prometheus Gauge:
sglang:weight_load_duration_seconds, 带 source 标签,
multiprocess_mode='mostrecent'。
2. 在 weight_updater.py 的 SchedulerWeightUpdaterManager 中添加可选字段
metrics_collector 和上下文管理器 _observe_weight_load, 用于测量耗时并调用
metrics_collector.observe_weight_load。
3. 将四个 update_weights_from_* 方法 (disk、distributed、tensor、ipc) 用该上下文管理器包裹, 以在调用结束时记录耗时和来源。
4. 在 scheduler.py 的 init_weight_updater 中将 self.metrics_collector 作为参数传入
SchedulerWeightUpdaterManager。

关键文件:

- python/sglang/srt/managers/scheduler_components/weight_updater.py (模块 权重更新器; 类别 source; 类型 core-logic; 符号 _observe_weight_load) : 核心变更文件, 添加上下文管理器包装四种权重更新路径的耗时计时。
- python/sglang/srt/observability/metrics_collector.py (模块 可观测性; 类别 source; 类型 core-logic; 符号 observe_weight_load, weight_load_duration_seconds) : 定义新

gauge 和 observe 方法

- python/sclang/srt/managers/scheduler.py (模块 调度器; 类别 source; 类型 configuration) : 传递 metrics_collector 到 weight_updater

关键符号: _observe_weight_load, observe_weight_load, init_weight_updater

关键源码片段

python/sclang/srt/managers/scheduler_components/weight_updater.py

核心变更文件, 添加上下文管理器包装四种权重更新路径的耗时计时。

```
from contextlib import contextmanager
from typing import Iterator, Optional
import time

@dataclass(kw_only=True, slots=True)
class SchedulerWeightUpdaterManager:
    # ... existing fields ...
    metrics_collector: Optional[Any] = None # 新增, 用于上报耗时指标

    @contextmanager
    def _observe_weight_load(self, source: str) -> Iterator[None]:
        # 边缘触发: 在 update_weights_from_* 调用结束时记录耗时。
        # engine 在更新期间暂停, 定时统计路径无法捕获此延迟。
        # `source` 取值 disk, distributed, tensor, ipc
        t0 = time.perf_counter()
        try:
            yield
        finally:
            if self.metrics_collector is not None:
                self.metrics_collector.observe_weight_load(
                    time.perf_counter() - t0, source
                )

    def update_weights_from_disk(self, recv_req):
        '''从磁盘原地更新权重。'''
        with self._observe_weight_load('disk'):
            success, message = self.tp_worker.update_weights_from_disk(recv_req)
            tp_success = success
            if success and self.draft_worker is not None:
                success, message = self.draft_worker.update_weights_from_disk(recv_req)
            if tp_success:
                self.flush_cache_after_weight_update(recv_req)
            if not success:
                logger.error(message)
            return UpdateWeightFromDiskReqOutput(success, message, 0)
```

python/sclang/srt/observability/metrics_collector.py

定义新 gauge 和 observe 方法

```
# =====
# Weight update
# =====
self.weight_load_duration_seconds = Gauge(
    name='sglang:weight_load_duration_seconds',
    documentation=(
        'Wall time of the most recent update_weights_from_<source> call on '
        'this scheduler rank (seconds). `source` label is one of: disk, '
        'distributed, tensor, ipc. Event-detection via '
        'changes(...[<range>]) > 0 — no separate counter needed.'
    ),
    labelnames=[*labels.keys(), 'source'],
    multiprocess_mode='mostrecent',
)

def observe_weight_load(self, duration_seconds: float, source: str) -> None:
    # 边缘触发: engine 在更新期间暂停, log_stats 不会触发,
    # 因此在 update_weights_from_* 末尾直接写入 gauge。
    # `source` 是 disk | distributed | tensor | ipc
    self.weight_load_duration_seconds.labels(**self.labels, source=source).set(
        duration_seconds
    )
```

评论区精华

无实质性技术讨论。Reviewerispobock直接批准。GeminiCodeAssist自动评论表示无反馈。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。仅在原有路径上增加计时和指标设置，未改变任何业务逻辑。
metrics_collector 可空，已有防御性检查（not None）。time.perf_counter() 开销极低，对性能无影响。
- 影响：对用户无直接行为影响，但提供新的监控维度。运维人员可通过
sglang:weight_load_duration_seconds 观察每个 rank 的各种权重来源耗时，辅助
tail-latency 分析和 dashboard 建设。
- 风险标记：低风险，可选依赖防御，无回归风险

关联脉络

- 暂无明显关联 PR