

PR #32363 完整报告

sgl-project/sglang

Add stream label to TTFT metrics

合并时间: 2026-07-25 08:44

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32363>

执行摘要

- 一句话: TTFT 指标按流模式拆分
- 推荐动作: 值得合并, 改动清晰且符合监控最佳实践。建议补充一个单元测试验证标签写入, 并在合并后通知运维团队更新 Dashboard 查询。

功能与动机

监控场景需要区分流式与非流式请求的首 Token 延迟, 以便更准确地分析性能异常。PR body 明确说明“Split `sglang:time_to_first_token_seconds` by streaming mode so dashboards can compare TTFT for streaming and non-streaming requests”。

实现拆解

1. `metrics_collector.py`: 直方图初始化时将 "stream" 加入 `labelnames`; 方法 `observe_time_to_first_token` 新增 `stream: bool` 参数, 调用时根据其值将 stream 标签设为 "true" 或 "false"。
2. `tokenizer_manager.py`: `collect_metrics` 在调用 `observe_time_to_first_token` 时, 通过 `getattr(state.obj, "stream", False)` 获取请求的流模式并传递。
3. `check_time_to_first_token_straggler` 中固定使用 `stream="true"` 的桶, 确保 straggler 检测只基于流式请求的分位数, 避免非流式请求的分布影响异常判定。

关键文件:

- `python/sglang/srt/observability/metrics_collector.py` (模块 可观测性; 类别 source; 类型 core-logic; 符号 `observe_time_to_first_token`): 核心变更文件: TTFT 直方图增加 stream 标签, `observe_time_to_first_token` 方法签名和实现修改, straggler 检测适配。
- `python/sglang/srt/managers/tokenizer_manager.py` (模块 调度器; 类别 source; 类型 core-logic): 调用方修改: 从请求对象获取 stream 标志并传递。

关键符号: `observe_time_to_first_token`, `check_time_to_first_token_straggler`, `collect_metrics`

关键源码片段

`python/sglang/srt/observability/metrics_collector.py`

核心变更文件：TTFT 直方图增加 stream 标签，`observe_time_to_first_token` 方法签名和实现修改，straggler 检测适配。

```
def __init__(self, ...):
    # ...
    self.histogram_time_to_first_token = Histogram(
        name="sglang:time_to_first_token_seconds",
        documentation="Histogram of time to first token in seconds.",
        # 新增 stream 标签，区分流式与非流式请求
        labelnames=[*labels.keys(), "stream"],
        buckets=bucket_time_to_first_token,
    )
    # ...

def observe_time_to_first_token(
    self, labels: Dict[str, str], value: float, *, stream: bool
):
    # 根据 stream 参数设置标签值 "true" 或 "false"
    self.histogram_time_to_first_token.labels(
        **labels, stream="true" if stream else "false"
    ).observe(value)

def check_time_to_first_token_straggler(self, value: float) -> bool:
    # straggler 检测仅基于流式请求的分位数
    his = self.histogram_time_to_first_token.labels(**self.labels, stream="true")
    total_observations = sum(bucket._value for bucket in his._buckets)
    if total_observations < 100:
        return False
    p99_threshold = total_observations * 0.99
    cumulative_count = 0
    for i, bucket in enumerate(his._buckets):
        cumulative_count += bucket._value
        if cumulative_count > p99_threshold:
            return value >= his._upper_bounds[i]
    return False
```

`python/sglang/srt/managers/tokenizer_manager.py`

调用方修改：从请求对象获取 stream 标志并传递。

```
def collect_metrics(self, state: ReqState, recv_obj: BatchStrOutput, i: int):
    # ...
    if not state.ttft_observed and self.disaggregation_mode != DisaggregationMode.PREFILL:
        state.ttft_observed = True
        state.last_completion_tokens = completion_tokens
        self.metrics_collector.observe_time_to_first_token(
            labels,
            state.time_stats.get_first_token_latency(),
            # 从请求对象获取 stream 标志，默认为 False
            stream=getattr(state.obj, "stream", False),
```

```
)  
# ...
```

评论区精华

无 Review 讨论。PR 提交后由 gemini-code-assist 发表了一个已弃用的自动评论，无实质内容。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 回归风险低：变更量小 (+12/-5)，仅追加一个标签和传递一个布尔值，逻辑清晰。
 2. straggler 行为变化：check_time_to_first_token_straggler 从使用无 stream 过滤的直方图改为固定 stream=true，若历史上 straggler 检测涵盖非流式请求，行为可能变窄，需确认是否预期。
 3. 缺少测试覆盖：PR 未新增对应测试用例，无法验证 stream 标签是否正确写入。 - 影响：影响范围小，仅修改可观测性模块。现有 Dashboard 若依赖原 sglang:time_to_first_token_seconds 无 stream 标签的时序，需更新 PromQL 以兼容新增标签（如加入 {stream="true"} 或 {stream="false"} 过滤）。 - 风险标记：缺少测试覆盖，straggler 行为变更

关联脉络

- 暂无明显关联 PR