PR #46768 完整报告

vllm-project/vllm

[Frontend] add per-request timing `metrics` field to response body of Chat/Completions APIs

合并时间：2026-07-07 08:48

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46768

执行摘要

- 一句话：Chat/Completions 响应体新增 per-request timing metrics 字段
- 推荐动作：建议精读 build_per_request_timing_metrics 函数以理解各时间段的计算边界（特别是 generation_time_ms vs tokens_per_second 的区别）。双门控设计值得参考，但团队可评估是否默认开启以简化用户配置。注意流式模式下对 include_usage 的依赖，需在文档中突出说明。

功能与动机

关联 Issue #40076 指出，vLLM 内部已追踪详细的 per-request 时序，但仅通过 Prometheus 或 OpenTelemetry 暴露聚合视图，API 消费者无法获取自身请求的精确时间分布。业务场景包括多租户计费、SLA 归因、延迟实验对比以及应用层调试。本 PR 将这些指标直接嵌入响应体，使客户端无需额外监控基础设施即可按请求精细分析性能。

实现拆解

1. 定义数据模型：在 vllm/entrypoints/openai/engine/protocol.py 中新增 PerRequestTimingMetrics Pydantic 模型，包含五个可空浮点字段。
2. 实现核心计算逻辑：在 vllm/entrypoints/generate/base/serving.py 中新增 build_per_request_timing_metrics 函数，接收 RequestStateStats 和生成 token 数，输出 PerRequestTimingMetrics。
3. 集成到 Chat 与 Completions serving 层：在 OpenAIServingChat 和 OpenAIServingCompletion 的 __init__ 中添加 enable_per_request_metrics 属性；在非流式响应构建方法和流式生成器中，当双重条件满足时调用 build_per_request_timing_metrics 并将结果赋值给响应对象的 metrics 字段。
4. 双门控与抑制规则：服务器级通过 CLI 参数 --enable-per-request-metrics 控制；请求级在 ChatCompletionRequest 和 CompletionRequest 中新增 include_metrics 字段（默认 False）。对于 n>1 和多 prompt 请求，metrics 为 null；若 --disable-log-stats 导致 RequestStateStats 不可用，各子字段为 null 并在启动时记录警告（最终版本改为 ValueError 阻止启动）。
5. 测试与文档：在 tests/entrypoints/openai/chat_completion/test_serving_chat.py 和 tests/entrypoints/openai/completion/test_completion_error.py 中添加约 25 个测试用例，覆盖指标计算、门控逻辑、流式附着等；新增文档 docs/features/per_request_metrics.

md。

关键文件：

- vllm/entrypoints/generate/base/serving.py（模块 指标计算；类别 source；类型 core-logic；符号 build_per_request_timing_metrics）：核心计算函数 build_per_request_timing_metrics 所在，定义了 RequestStateStats 到 PerRequestTimingMetrics 的转换逻辑。
- vllm/entrypoints/openai/engine/protocol.py（模块 协议模型；类别 source；类型 core-logic；符号 PerRequestTimingMetrics）：定义了 PerRequestTimingMetrics Pydantic 模型，作为 API 响应数据结构。
- vllm/entrypoints/openai/chat_completion/serving.py（模块 Chat 服务；类别 source；类型 core-logic）：Chat serving 层集成 metrics，在流式和非流式响应中注入 metrics 字段。
- vllm/entrypoints/openai/completion/serving.py（模块 Completion 服务；类别 source；类型 core-logic）：Completions serving 层集成 metrics，类似 Chat 层。
- vllm/entrypoints/openai/cli_args.py（模块 CLI 参数；类别 source；类型 configuration）：添加 --enable-per-request-metrics 命令行参数，并实现与 --disable-log-stats 的冲突检测。
- docs/features/per_request_metrics.md（模块 文档；类别 docs；类型 documentation）：新增功能文档，描述用法和限制。

关键符号：build_per_request_timing_metrics, PerRequestTimingMetrics

关键源码片段

vllm/entrypoints/generate/base/serving.py

核心计算函数 build_per_request_timing_metrics 所在，定义了 RequestStateStats 到 PerRequestTimingMetrics 的转换逻辑。

```
def build_per_request_timing_metrics(
    metrics: RequestStateStats | None,
    num_generation_tokens: int,
) -> PerRequestTimingMetrics:
    # 从 RequestStateStats 构建 per-request 时序指标。
    # generation_time_ms 是纯解码间隔（首 token 到尾 token），
    # 不包含队列等待和 prefill/TTFT。
    # tokens_per_second 是整体输出吞吐量，包含 prefill 阶段，
    # 因此不总是 mean_itl_ms 的倒数。
    if metrics is None:
        return PerRequestTimingMetrics()
    queued_ts = metrics.queued_ts
    scheduled_ts = metrics.scheduled_ts
    first_token_ts = metrics.first_token_ts
    last_token_ts = metrics.last_token_ts
    time_to_first_token_ms: float | None = None
    generation_time_ms: float | None = None
    queue_time_ms: float | None = None
```

```
    mean_itl_ms: float | None = None
    tokens_per_second: float | None = None
    # TTFT：从调度到首 token 的时间
    if scheduled_ts > 0 and first_token_ts > 0:
        time_to_first_token_ms = (first_token_ts - scheduled_ts) * 1000
    # 生成时间：首 token 到尾 token
    if first_token_ts > 0 and last_token_ts > 0:
        generation_time_ms = (last_token_ts - first_token_ts) * 1000
    # 队列等待时间：入队到调度
    if queued_ts > 0 and scheduled_ts > 0:
        queue_time_ms = (scheduled_ts - queued_ts) * 1000
    # 平均 inter-token 延迟：仅当多于 1 个生成 token 时计算
    if first_token_ts > 0 and last_token_ts > 0 and num_generation_tokens > 1:
        decode_time = last_token_ts - first_token_ts
        mean_itl_ms = decode_time / (num_generation_tokens - 1) * 1000
    # 整体 token 吞吐率：生成 token 数 / 推理时间（调度到结束）
    if scheduled_ts > 0 and last_token_ts > 0:
        inference_time_ms = (last_token_ts - scheduled_ts) * 1000
        if inference_time_ms > 0:
            tokens_per_second = num_generation_tokens / inference_time_ms * 1000
    return PerRequestTimingMetrics(
        time_to_first_token_ms=time_to_first_token_ms,
        generation_time_ms=generation_time_ms,
        queue_time_ms=queue_time_ms,
        mean_itl_ms=mean_itl_ms,
        tokens_per_second=tokens_per_second,
    )
```

vllm/entrypoints/openai/engine/protocol.py

定义了 PerRequestTimingMetrics Pydantic 模型，作为 API 响应数据结构。

```
class PerRequestTimingMetrics(OpenAIBaseModel):
    # 每个请求的时序指标，单位均为毫秒（tokens_per_second 除外）。
    time_to_first_token_ms: float | None = None # 首 token 延迟 (TTFT)
    generation_time_ms: float | None = None # 纯生成时间（解码阶段）
    queue_time_ms: float | None = None # 在调度队列中的等待时间
    mean_itl_ms: float | None = None # 平均 token 间延迟
    tokens_per_second: float | None = None # 整体输出吞吐率（含 prefill）
```

评论区精华

Review 中 simon-mo 提出了三点意见：

- CLI 冲突处理：当 --enable-per-request-metrics 与 --disable-log-stats 同时设置时应 raise ValueError 而非仅记录警告，最终实现改为 ValueError。
- 默认值偏好：倾向默认开启但接受当前关闭，未来可重新评估。
- 请求级 include_metrics 必要性：质疑是否应仅引擎层控制，但作者保留请求级参数以提供灵活性。

- CLI 参数冲突处理方式 (design): 开发者将警告改为 ValueError（通过 commit 'chore: moving warning to post-parse CLI arg validator' 实现）。
- 默认开启 vs 当前关闭 (design): 保留默认关闭，后续可重新评估。
- 请求级别 include_metrics 参数的必要性 (design): 作者坚持保留请求级开关，认为提供灵活性有价值，最终未移除。

风险与影响

- 风险:
  - 数据来源依赖：指标依赖引擎 RequestStateStats，若用户禁用日志统计，所有字段为 null。当前版本已阻止不兼容组合启动（raise ValueError）。
  - 流式附着机制：流式模式要求客户端同时设置 include_metrics: true 和 stream_options.include_usage: true，否则无法收到指标，此约束可能不被用户察觉。
  - 归属歧义抑制：对于 n>1 和多 prompt 请求，metrics 直接返回 null，功能降级透明但可能意外。
  - 计算开销：每次响应增加若干浮点运算和一次对象构造，影响可忽略。
- 影响:
  - 用户侧：API 消费者现可在单次请求响应中获得完整时序分解，方便计费、SLA 监控和调试。
  - 系统侧：响应体增加额外字段，序列化开销极小，仅在用户启用时生效。
  - 团队侧：新增配置项和测试套件，需维护兼容性。与 PR #42198（header-based 指标）互补，提供两种暴露途径。
  - 风险标记：数据依赖日志，流式需 usage chunk, 归属歧义抑制，计算开销极低

关联脉络

- PR #40076 [Feature]: Per-request timing metrics in response body: 功能需求 issue，PR 直接解决该需求。
- PR #42198 [Per-request metrics via response headers]: 类似功能但通过 HTTP headers 暴露，两个 PR 互补，提供不同消费方式。