PR #24000 完整报告

sgl-project/sglang

[tokenizer] Surface scheduler load info (num_running_reqs / num_waiting_reqs) in meta_info

合并时间：2026-06-01 11:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24000

执行摘要

- 一句话：在 meta_info 中暴露调度器负载信息
- 推荐动作：值得合入。这个小而清晰的变更充分利用了已有数据通道，消除了冗余的负载轮询。推荐阅读 _handle_batch_output 方法以理解数据流路径。

功能与动机

在 router 风格的部署中，客户端需要基于调度器队列长度进行响应式流控（如节流）。此前，调度器已在每个 BatchStrOutput / BatchTokenIDOutput 中通过 recv_obj.load 字段携带了每 DP rank 的负载指标，但 TokenizerManager 仅将其用于 DP 均衡的 WatchLoadUpdateReq 路径，未在 meta_info 中暴露给用户。这迫使客户端每间隔轮询 /v1/loads，给调度器增加了额外负担。

实现拆解

1. 定位变更入口：在 python/sglang/srt/managers/tokenizer_manager.py 的 _handle_batch_output 方法中，在构建 meta_info 字典之后、处理其他逻辑之前插入代码。
2. 安全地提取负载信息：使用 getattr(recv_obj, "load", None) 获取负载对象，避免因 recv_obj 类型（如 BatchEmbeddingOutput）不携带该字段而报错。
3. 转发字段：从 load 对象中安全提取 num_running_reqs 和 num_waiting_reqs，若存在则插入 meta_info。由于该字段是每个请求共享的同一对象引用，因此开销极低。
4. 无配置变更：不引入新的服务参数或环境变量，行为完全向后兼容——现有客户端忽略新增字段。

关键文件：

- python/sglang/srt/managers/tokenizer_manager.py（模块 请求路由；类别 source；类型 core-logic；符号 _handle_batch_output）：唯一变更文件，核心入口 _handle_batch_output 方法中插入 13 行代码，将调度器负载信息暴露到 meta_info 中。

关键符号：_handle_batch_output

关键源码片段

python/sglang/srt/managers/tokenizer_manager.py

唯一变更文件，核心入口 _handle_batch_output 方法中插入 13 行代码，将调度器负载信息暴露到 meta_info 中。

```
# 位于 _handle_batch_output 方法中，meta_info 字典初始化之后
# 从 recv_obj 获取调度器负载信息（若存在）
load = getattr(recv_obj, "load", None)
if load is not None:
    # 使用 getattr 安全提取字段，避免 load 对象版本不一致
    num_running_reqs = getattr(load, "num_running_reqs", None)
    num_waiting_reqs = getattr(load, "num_waiting_reqs", None)
    # 仅当字段不为 None 时插入，保持向后兼容
    if num_running_reqs is not None:
        meta_info["num_running_reqs"] = num_running_reqs
    if num_waiting_reqs is not None:
        meta_info["num_waiting_reqs"] = num_waiting_reqs
```

评论区精华

Review 讨论较少，仅有 reviewer JustinTong0323 批准，认为“这是一个小的累加暴露，现有守卫机制保证了非生成路径的安全”。未出现设计争议或未解决的问题。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅添加了条件性字典插入，使用 getattr 确保对所有输出类型安全。不影响任何核心路径，不会引发回归。但如果下游客户端依赖 meta_info 的精确键集合，新增字段理论上可能破坏严格的 schema 验证（如使用 Pydantic 且禁止额外字段），但这属于预期行为。
- 影响：对用户：对于使用 SGLang SDK 或直接解析 meta_info 的客户端，现在可以直接通过 num_running_reqs 和 num_waiting_reqs 字段感知调度器负载，便于实现响应式流控，减少轮询开销。对系统：无性能影响，数据已存在于内存中。对团队：无后续维护负担。
- 风险标记：数据依赖，无测试覆盖

关联脉络

- 暂无明显关联 PR