PR #28408 完整报告

sgl-project/sglang

Remove stale load collection from output streaming hot path

合并时间：2026-06-18 06:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28408

执行摘要

- 一句话：移除输出热路径中过时的负载收集
- 推荐动作：建议在合并前优先解决 WatchLoadUpdateReq 被外部包导入的兼容性问题，可以为 smg-grpc-servicer 提供兼容性 shim 或标记为弃用。PR 本身逻辑正确，但缺少测试覆盖，建议补充单元测试验证移除后不影响核心路径。值得关注的设计决策是清理旧 API 时需充分评估外部依赖。

功能与动机

PR #26348 已将 DP 负载更新和 /v1/loads 端点迁移到共享内存负载快照，但输出流热路径仍调用 get_loads(include=["core"]) 并将结果附加到 BatchTokenIDOutput.load 中。此 piggybacked load 已不再被主负载路径使用，因此可以安全移除，以简化热路径并减少不必要的开销。

实现拆解

1. 删除 BatchTokenIDOutput.load 和 BatchStrOutput.load 字段：在 io_struct.py 中移除这两个数据类的 load: GetLoadsReqOutput = None 字段定义，消除响应结构中的负载携带。
2. 移除 _stream_output_generation 中的负载查询：在 output_streamer.py 中删除对 self.load_inquirer_get_loads(GetLoadsReqInput(include=["core"])) 的调用，以及将结果传递给 acc.to_payload(load=load) 的逻辑。方法签名也从 to_payload 中移除 load 参数。
3. 移除 TokenizerManager._handle_batch_output 中的负载附加逻辑：在 tokenizer_manager.py 中删除从 recv_obj.load 提取 num_running_reqs 和 num_waiting_reqs 并注入到响应 meta_info 的代码块。
4. 删除 WatchLoadUpdateReq 数据类：在 io_struct.py 中移除整个 WatchLoadUpdateReq 定义，因为其不再被使用。
5. 清理 SchedulerOutputStreamer 和 Scheduler 中的负载相关注入：从 output_streamer.py 的类定义中移除 load_inquirer_get_loads 字段，并从 scheduler.py 的 init_output_streamer 中移除对应的 lambda 传参。

关键文件：

- python/sglang/srt/managers/io_struct.py（模块 数据结构；类别 source；类型 core-logic；符号 WatchLoadUpdateReq）：移除了 BatchTokenIDOutput.load、

BatchStrOutput.load 字段和整个 WatchLoadUpdateReq 数据类，是清理的核心所在。

- python/sglang/srt/managers/tokenizer_manager.py（模块 Token 管理器；类别 source；类型 core-logic）：移除了从响应 meta_info 中提取负载信息的逻辑块，减少每个请求处理路径上的开销。
- python/sglang/srt/managers/scheduler_components/output_streamer.py（模块 输出流；类别 source；类型 core-logic）：删除了从 output_streamer 类中调用 load_inquirer_get_loads 的代码，并移除了 to_payload 方法的 load 参数，这是清理热路径负载收集的关键。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：移除 init_output_streamer 中 load_inquirer_get_loads 的 lambda 传参，是与 output_streamer 配套的清理。

关键符号：_stream_output_generation, _handle_batch_output, to_payload

关键源码片段

python/sglang/srt/managers/io_struct.py

移除了 BatchTokenIDOutput.load、BatchStrOutput.load 字段和整个 WatchLoadUpdateReq 数据类，是清理的核心所在。

```
# python/sglang/srt/managers/io_struct.py

# 之前：
# @dataclass
# class BatchTokenIDOutput:
# ...
# load: GetLoadsReqOutput = None # 已移除

# 之前：
# @dataclass
# class BatchStrOutput:
# ...
# load: GetLoadsReqOutput = None # 已移除

# 之前：
# @dataclass
# class WatchLoadUpdateReq(BaseReq):
# loads: List[GetLoadsReqOutput]
# 已完全移除，但被 smg-grpc-servicer 外部包依赖
```

python/sglang/srt/managers/tokenizer_manager.py

移除了从响应 meta_info 中提取负载信息的逻辑块，减少每个请求处理路径上的开销。

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

async def _handle_batch_output(self, recv_obj):
    for i, rid in enumerate(recv_obj.rids):
        state = self.rid_to_state.get(rid, None)
```

```
        # ...
        meta_info = {
            "id": rid,
            "finish_reason": recv_obj.finished_reasons[i],
            "prompt_tokens": recv_obj.prompt_tokens[i],
            "weight_version": self.server_args.weight_version,
            "num_retractions": recv_obj.retraction_counts[i],
        }

        # 以下代码块已被移除：
        # load = getattr(recv_obj, "load", None)
        # if load is not None:
        # num_running_reqs = getattr(load, "num_running_reqs", None)
        # num_waiting_reqs = getattr(load, "num_waiting_reqs", None)
        # 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

        if self.enable_metrics:
            if recv_obj.time_stats is not None:
                scheduler_time_stats = recv_obj.time_stats[i]
                meta_info.update(scheduler_time_stats.convert_to_output_meta_info())
        # ... 继续处理
```

python/sglang/srt/managers/scheduler_components/output_streamer.py

删除了从 output_streamer 类中调用 load_inquirer_get_loads 的代码，并移除了 to_payload 方法的 load 参数，这是清理热路径负载收集的关键。

```
# python/sglang/srt/managers/scheduler_components/output_streamer.py

@dataclass(kw_only=True, slots=True)
class SchedulerOutputStreamer:
    send_to_detokenizer: zmq.Socket
    tree_cache: BasePrefixCache
    ps: ParallelState
    server_args: ServerArgs
    is_generation: bool
    spec_algorithm: SpeculativeAlgorithm
    disaggregation_mode: DisaggregationMode
    enable_hicache_storage: Callable[[], bool]
    # load_inquirer_get_loads: Callable[..., Any] # 已移除
    _test_stream_output_count: int = 0

    def _stream_output_generation(self, reqs, ...):
        # ... 构造 acc ...
        # load = self.load_inquirer_get_loads(GetLoadsReqInput(include=["core"])) # 已移除
        for req in reqs:
            # ...
```

```
payload = acc.to_payload(
    # load=load, # 已移除
    dp_rank=self.ps.dp_rank,
    is_idle_batch=is_idle_batch,
    has_reqs=bool(reqs),
)
```

评论区精华

评审中，gemini-code-assist[bot] 自动评论确认无额外反馈。ShangmingCai 批准了变更，并邀请作者协助审查 PR #26561，在 PD 分解设置下联合推进解码端负载均衡。@junliu-mde 在 Issue 评论中指出，移除 WatchLoadUpdateReq 会导致 gRPC 模式下 smg-grpc-servicer 依赖该符号的导入失败（ImportError: cannot import name 'WatchLoadUpdateReq' from 'sglang.srt.managers.io_struct'）。这一兼容性问题在 PR 合并前未被解决。

- WatchLoadUpdateReq 被外部包依赖导致 gRPC 模式不可用 (correctness): 未在 PR 合并前解决，这是一个已识别的回归 bug。

风险与影响

- 风险：
  1. gRPC 模式兼容性风险：WatchLoadUpdateReq 被 smg-grpc-servicer 包导入，PR 直接删除该符号会导致 gRPC 模式无法使用（已由 @junliu-mde 在 Issue 中确认）。这是一个回归性 bug，影响在生产环境中使用 gRPC 接口的用户。
  2. 缺少测试覆盖：PR 未对移除的负载携带逻辑添加任何适配性测试，移除后原有依赖该负载信息的客户端可能遇到 meta_info 中缺少 num_running_reqs / num_waiting_reqs 字段的问题，但 PR 未提供迁移路径。 - 影响：影响范围：中等。移除的代码位于输出流热路径和 TokenizerManager 中，涉及请求响应的构建流程。影响程度：高。对于使用 gRPC 模式的用户，该 PR 直接导致 smg-grpc-servicer 导入失败，功能不可用；对于 HTTP 接口，num_running_reqs / num_waiting_reqs 字段将从响应中消失，依赖这些字段进行客户端流量控制的用户需迁移到 /v1/loads 端点。影响团队：维护 smg-grpc-servicer 的团队需要更新依赖适配移除的符号；所有依赖响应负载字段的客户端团队需要调整。
- 风险标记：缺少测试覆盖，外部依赖破坏，核心路径变更

关联脉络

- PR #26348 Move DP load updates and /v1/loads to shared-memory load snapshots: 本 PR 是 #26348 的后续清理，移除其迁移后过时的 load 携带代码。
- PR #24000 DP balancing and /v1/loads continue to read from load snapshots: 引入了早期的负载 piggyback 机制，本 PR 移除了该机制过时的部分。