

PR #2028 完整报告

THUDM/slime

remove abundant function

合并时间: 2026-06-06 23:03

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2028>

执行摘要

- 一句话: 移除未使用的 `calculate_workload` 函数
- 推荐动作: 建议快速合并。清理死代码是良好的工程实践, 该 PR 本身低风险、价值明确。
值得关注的设计决策: 无。

功能与动机

PR body 和 commit message 仅提及 "remove abundant function", 未提供更详细背景。该函数 `calculate_workload` 在仓库现有代码中已无调用方 (经源码搜索确认), 属于死代码清理, 旨在减少维护负担。

实现拆解

1. 在 `slime/utils/seqlen_balancing.py` 中删除了 `calculate_workload` 函数定义 (第 180-189 行原内容)。该函数接受 `seqlen_list` 和 `coeff` 参数, 返回按 $\text{coeff} * L + L^2$ 计算的 FLOPs 估计值。
2. 删除了该函数的完整 docstring 和类型处理逻辑 (对 list 返回 list, 对标量返回标量)。
3. 未修改其他文件或引用点, 因为该函数已被确定在任何模块中均未使用 (依赖分析确认)。

关键文件:

- `slime/utils/seqlen_balancing.py` (模块 序列平衡; 类别 source; 类型 core-logic; 符号 `calculate_workload`): 唯一修改的文件, 删除了 `calculate_workload` 函数, 清理死代码。

关键符号: `calculate_workload`

关键源码片段

`slime/utils/seqlen_balancing.py`

唯一修改的文件, 删除了 `calculate_workload` 函数, 清理死代码。

```
# slime/utils/seqlen_balancing.py (移除前)
# 原 calculate_workload 函数已被删除, 下为删除前的完整定义:
# def calculate_workload(seqlen_list, coeff=24576):
#     """Estimate per-sequence transformer FLOPs proportional to coeff * L + L^2.
#     # Pre-computed from model config as 2*h + ffn_mul * d_ff // 2.
#     # Default 24576 = 6*4096."""
```

```
# if isinstance(seqlen_list, list):
# return [coeff * sl + sl * sl for sl in seqlen_list]
# return coeff * seqlen_list + seqlen_list**2
```

评论区精华

无 review 评论或讨论，属作者自行清理。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：删除的仅为明确未使用的函数，不影响任何现有功能或接口。潜在的微小风险是若有外部用户依赖该函数（通过导入或 monkey-patch），但作为内部工具函数，其接口不稳定且无公开文档保证，可视为安全。
- 影响：对用户和系统无运行时影响。对团队而言，代码库减少了死代码，提升了可读性和可维护性。对未来的影响：若后续需要类似 FLOPs 估算功能，需重新实现或从 git 历史中恢复。
- 风险标记：低风险

关联脉络

- PR #2029 use balance_by_flops as balance data across mbs: PR#2029 引入了基于 FLOPs 的平衡调度，与 calculate_workload 的 FLOPs 估算目标相关，但采用了不同的实现方式（balance_by_flops），可能使得 calculate_workload 成为冗余，进而被此 PR 清理。