

# PR #2029 完整报告

THUDM/slime

use balance\_by\_flops as balance data across mbs

合并时间: 2026-06-06 23:51

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2029>

## 执行摘要

- 一句话: balance\_by\_flops 全面接管平衡调度
- 推荐动作: 推荐阅读 dp\_schedule.py 的调度逻辑变更, 该 PR 体现了实际 FLOPs 建模优于简单 token 计数的设计思路, 对性能敏感训练有意义。

## 功能与动机

在长序列训练场景下, token 数不能准确反映计算量, FLOPs 估算能更均衡地分配 workload。PR body 中明确指出了将 balance\_by\_flops 用作跨 mbs 数据平衡的依据。

## 实现拆解

1. 统一平衡策略: 在 build\_dp\_schedule 中, 去除了旧的 balance\_by\_flops 条件分支, 使 balance\_data 始终使用 FLOPs 权重 (通过 \_calculate\_workloads) 进行 KK 分配, 而非 token 计数。
2. 简化分发逻辑: 删除 slime/utils/dp\_schedule.py 中 mbs\_weights 按 token 计算的替代路径, 所有 balance\_data=True 时均基于 FLOPs 权重。
3. 配置联动: slime/utils/arguments.py 中, 当设置 balance\_by\_flops 时自动启用 balance\_data, 实现两种模式的无缝衔接。
4. 文档更新: 更新了 --balance-data 和 --balance-by-flops 的帮助描述, 明确后者可能使 mbs 超出 token cap 导致 OOM。
5. 测试补充: 在 tests/test\_dp\_schedule.py 中新增 test\_balance\_data\_distributes\_by\_flops, 验证 FLOPs 权重下的分区结果正确性。

关键文件:

- slime/utils/dp\_schedule.py (模块 调度器; 类别 source; 类型 core-logic) : 核心调度逻辑变更, 统一了跨 rank 分发使用 FLOPs 权重
- tests/test\_dp\_schedule.py (模块 调度器; 类别 test; 类型 test-coverage; 符号 test\_balance\_data\_distributes\_by\_flops) : 新增测试覆盖 balance\_data 使用 FLOPs 权重的场景
- slime/utils/arguments.py (模块 配置; 类别 source; 类型 configuration) : 更新参数描述并添加 balance\_by\_flops 自动启用 balance\_data 的逻辑

关键符号: `_pack_step_into_mbs`, `build_dp_schedule`,  
`test_balance_data_distributes_by_flops`

## 关键源码片段

### `slime/utils/dp_schedule.py`

核心调度逻辑变更, 统一了跨 rank 分发使用 FLOPs 权重

```
# slime/utils/dp_schedule.py (build_dp_schedule 节选)
# 在包好 micro-batch 后, 将 mbs 分发到 DP rank
# 注意: 现在总是使用 FLOPs 权重, 而非 token 计数
step_workloads = _calculate_workloads(step_lengths, args)
mbs_weights = [sum(step_workloads[i] for i in bin_) for bin_ in step_mbs]
rank_mbs_idx = get_seqlen_balanced_partitions(mbs_weights, dp_size, equal_size=True)
```

### `tests/test_dp_schedule.py`

新增测试覆盖 `balance_data` 使用 FLOPs 权重的场景

```
# tests/test_dp_schedule.py
@pytest.mark.unit
def test_balance_data_distributes_by_flops():
    """balance_data 使用 FLOPs 权重进行 rank 分配, 而非原始 token 总数。"""
    total_lengths = [1, 2, 3, 4, 5, 7, 9, 10]
    rollout_indices = list(range(8))
    args = make_args(micro_batch_size=1, balance_data=True)
    tp = make_tp(dp_size=2)
    partitions, mbi, nmb, gbs_per_step = build_dp_schedule(
        args, tp, total_lengths, global_batch_size=8, rollout_indices=rollout_indices
    )
    # 验证分区结果符合 FLOPs 权重预期
    assert partitions == [[1, 2, 5, 6], [0, 3, 4, 7]]
    assert nmb == [4]
    assert gbs_per_step == [8]
    assert_invariants(partitions, mbi, nmb, dp_size=2,
                      expected_global_sample_indices=range(8),
                      total_lengths=total_lengths)
```

## 评论区精华

该 PR 无 review 评论, 提交者自行合并。从 commit 信息可推断核心决策是: 让 `balance_by_flops` 同时控制 mbs 内部 packing 和跨 rank 分发, 以统一调度语义。

- 暂无高价值评论线程

## 风险与影响

- 风险:
  1. 回归风险: `balance_data` 不再支持 token 计数分发, 若用户之前依赖该行为, 升级后 workload 分布可能变化。但 `balance_by_flops` 默认关闭, 默认行为不变。

2. OOM 风险: `balance_by_flops` 的 packing 不保证 token cap, 文档已警告, 但用户若不加注意仍可能 OOM。
3. 兼容性: `balance_by_flops` 自动启用 `balance_data`, 与旧配置混合使用时可能导致预期外的行为, 但 check 逻辑会确保正确性。 - 影响: 影响范围限于 DP 调度模块, 对用户透明 (默认行为不变)。开启 `--balance-by-flops` 的用户将获得更精确的 workload 均衡, 提升训练效率。 - 风险标记: 核心路径变更, 测试覆盖中等, OOM 风险提示

## 关联脉络

- PR #2013 Revert "rename rollout\_ids to group\_ids": 修改了相同文件 `dp_schedule.py`, 是调度逻辑的另一项变更