

slime 周报 #21: 架构重构与性能优化 (2026-05-18 ~ 2026-05-24)

THUDM/slme

周期: 2026-05-18 至 2026-05-24

来源 PR: 9 • 重点 PR: 9 • 自动生成

原文链接: <http://prhub.com.cn/THUDM/slme/reports/2026-05-18-to-2026-05-24>

执行摘要

本周 (2026-05-18 至 2026-05-24), THUDM/slme 仓库经历了显著的架构演进与性能优化。9 个重点 PR 中, 大部分由核心贡献者 zhuzilin 主导, 围绕训练调度、内存优化和异步 rollout 展开。整体趋势是将解耦和全局优化理念从实验性代码推向主线, 为更高效的分布式训练奠定基础。团队本周应重点关注调度迁移和可变 batch size 引入对现有训练管线的潜在影响, 并加强对新模块的测试覆盖。

本周重点变化

1. 调度逻辑迁移: PR#1926 将微批次调度 (micro-batch scheduling) 从训练侧的 DataIterator 迁移至 rollout 侧的 `_split_train_data_by_dp`, 利用全局信息进行 first-fit bin packing, 消除各 DP rank 间的概率不平衡, 并移除冗余 all-reduce 同步。这是继 PR#1930 可变 batch size 后的核心架构调整。
2. 动态全局 Batch Size: PR#1930 引入 `global_batch_sizes` 数组, 允许训练过程中每个 step 使用不同大小的全局 batch size, 不再依赖固定配置。该改动涉及 loss scaling 和 LR 调度器参数传递链路, 为 uneven DP 分区提供了基础。
3. 异步与流式 Rollout 正式入库: PR#1920 将实验性的 fully async rollout 从 examples 迁移至 `slime/rollout/`, 通过后台线程持续消费 `data_buffer`, 打破训练与 rollout 的强同步依赖。PR#1921 新增 `sglang` 流式生成函数 `generate_streaming`, 支持在 abort 时保留部分 token, 减少资源浪费。
4. 主机内存优化组合拳: PR#1924 升级 `torch_memory_saver`, 集成零拷贝 CPU 备份和梯度缓冲区禁用功能; PR#1932 调换配置项赋值顺序, 确保 critic 模式下 offload 正确生效。两项改动协同将主机内存消耗降低。
5. 可观测性增强: PR#1919 为 critic 角色添加独立 W&B 配置记录, 便于区分 actor/critic 的 CLI 参数, 但配置键前缀 (`critic/`) 可能影响现有面板, 需提前同步。

模块与主题趋势

本周标签统计显示 performance (6 次) 和 feature (4 次) 占据主导, 其次是 architecture 和 test (各 3 次), 印证了“优化 + 基建”的基调。热点文件集中在 `slime/ray/rollout.py`、`slime/backends/megatron_utils/data.py` 和 `slime/utils/arguments.py`, 均为训练数据流和参数校验的核心模块。新的 `slime/utils/dp_schedule.py` 从 Ray 依赖中解耦出来, 体现了团队对纯函数可测试性的追求。作者分布上, zhuzilin 贡献了 67% 的 PR, 其工作集中在调度、异步、

流式和内存优化四条线索，反映出该角色在架构层面的主导地位。

风险观察

- 核心路径变更风险：PR#1926 和 PR#1930 训练调度与数据传递链路深度重构，虽然代码设计清晰，但若回归测试不充分，可能导致训练结果偏差或出现隐藏的同步死锁。
- 新模块成熟度：fully async rollout (PR#1920) 和 streaming rollout (PR#1921) 作为新功能，尚未配套完善的单元测试（前者有集成测试，后者依赖 sglang SSE 累积语义），在各类环境下的稳定性有待验证。
- 配置传递变更：PR#1930 中 loss 和 LR 参数改用 global_batch_sizes 而非 args.global_batch_size，下游自定义训练循环若未同步适配，会引发运行时错误。
- 依赖升级无验证：PR#1916 仅更新了 torch_memory_saver 的 commit SHA，未附带功能验证或兼容性测试，一旦上游引入破坏性修改，将影响整个 Docker 部署环境。
- W&B 面板影响：PR#1919 为 critic 配置增加 critic/ 前缀，现有仪表盘若按原始键名构建，将丢失数据，需提醒团队更新面板配置。

重点 PR 速览

- PR#1926: Move micro-batch scheduling to rollout side。将调度逻辑从 DataIterator 中剥离，通过 slime/utils/dp_schedule.py 实现纯函数式调度，配套 test_dp_schedule.py 验证 bin packing 不变量。这是本周最重要架构调整。
- PR#1930: [1/N] Support training with variable global batch size。引入 global_batch_sizes 数组，修改 train_one_step 和 gather/log 函数，为后续 uneven-DP 分区铺垫。
- PR#1920: Move fully_async example to main codebase。将 AsyncRolloutWorker 迁入 slime/rollout/，并增加 CI 测试，使异步 rollout 成为官方支持特性。
- PR#1921: Add example for streaming output。新增 generate_streaming 函数，利用 SSE 流式解析实现增量状态累积，减少 abort 时的资源损失。
- PR#1924/1932: 内存优化组合。升级 torch_memory_saver 并加上配置顺序修复，有效降低 CPU 端内存峰值，适合大规模训练场景。
- PR#1919: Add critic wandb config。为 critic 角色添加独立配置记录，提升实验可观测性，但需注意前后端同步。

后续建议

- 加强回归测试：针对本周两条核心路径变更（PR#1926、PR#1930），建议扩展 tests/test_dp_schedule.py 覆盖更多边界用例（如单样本 step、极端变长序列），并在 CI 中增加与现有训练脚本的集成测试。
- 新模块监控：fully async 和 streaming rollout 应先在测试环境中运行 PPO 短作业，观察并发稳定性与内存走势，再推广到生产。
- 依赖锁定：建议将 torch_memory_saver commit 固化并附加版本对比测试，避免意外破坏。
- 文档同步：更新配置文档和 W&B 面板指南，体现 critic/ 前缀变更，减少团队困惑。

- 关注后续系列：zhuzilin 标记了 $[1/N]$ 的可变 batch size 系列，后续 PR 可能涉及 uneven-DP 正式实现，建议提前 review 设计文档。