

THUDM/slime 2026 年第 28 周周报 (07-06 至 07-12)

THUDM/slime

周期: 2026-07-06 至 2026-07-12

来源 PR: 5 • 重点 PR: 5 • 自动生成

原文链接: <http://prhub.com.cn/THUDM/slime/reports/2026-07-06-to-2026-07-12>

执行摘要

本周共合并 5 个 PR，整体聚焦于基础设施重构与功能增强。主要变化包括：引擎侧权重同步架构重构 (/pull_weights)、新增训练资源释放机制 (--release-train)、过时代码清理、routed experts 扩展以及数据来源追踪。作者分布以 zhuzilin 为主 (4 个)，nanjiangwill 贡献了高重要性的引擎重构。风险方面，测试覆盖不足仍是主要关注点，核心路径变更需后续验证。

本周重点变化

- 引擎权重同步下沉: PR #2181 实现了引擎侧 /pull_weights 端点，将磁盘级 delta 权重同步的 fan-out 从 slime 层下移到引擎内部，slime 不再需要了解引擎拓扑。该变更涉及 slang_engine.py、disk_delta.py 和补丁文件，对非 POSIX 文件系统兼容性做了处理。
- 训练资源释放模式: PR #2180 引入 --release-train 参数，在模型保存后释放训练 Actor 组的 GPU 资源，下一轮训练时从 checkpoint 重建。该模式与磁盘权重更新配合，可提高大规模集群的 GPU 利用率。涉及 actor_group.py、placement_group.py 和 train.py 的核心重构。
- 代码清理与多节点安全强化: PR #2183 删除了多个过时函数 (get_advantages_and_returns、onload_weights_from_disk 等)，移除了冗余参数，并增加了多节点环境下的 guard，提升代码健壮性。
- routed experts 支持: PR #2185 为 Sample 类添加 routed_experts_start_len，允许在解码阶段分块接收 routed experts 信息，同时放宽 log probs 复用条件。
- 数据来源追踪: PR #2184 新增 source_names 字段，通过 get_source 函数提取样本来源，支持多数据源混合训练场景。

模块与主题趋势

本周变更集中在以下模块:

- backends/slang_utils: 引擎侧新增 /pull_weights 端点，是该模块最重要的重构。slang_engine.py 修改次数最多。
- ray/rollout.py: 涉及多个 PR，包括数据来源追踪和代码清理，是数据流的核心路径。
- ray/actor_group.py: release-train 模式的重构，涉及资源生命周期管理。
- backends/megatron_utils: routed experts 支持，影响 actor.py 和 types.py。

- `utils/disk_delta.py`: 引擎重构中用于 `delta` 应用的模块。
- 配置与安全: 新增配置项 (`--release-train`) 和参数清理, 体现了团队在代码质量和安全上的持续投入。

趋势上, 本周表现出以下特点:

- 架构下沉: 将逻辑从 `orchestrator` 层下移至引擎层 (#2181), 减少中间层的耦合。
- 资源弹性: 通过释放和重建 Actor 组, 向弹性训练迈进一步。
- 债务清理: 系统性地移除过时代码, 降低维护成本。
- 功能扩展: 支持更多 MoE 路由模式和混合数据源。

风险观察

- 测试覆盖不足: 多个 PR (#2185、#2181、#2184) 均标记“缺少测试覆盖”, 尤其是路由变更和引擎重构无配套测试, 存在回归风险。
- 核心路径变更: `routed_experts_start_len` 涉及数据拼接逻辑, `release-train` 改变训练流程, 需在更多场景验证。
- 跨模块耦合: 引擎补丁与 `sglang` 后端强绑定, 未来切换后端可能需要调整。
- 删除函数残留: 清理 PR 删除了多个函数, 需确认所有依赖模块已适配, 尤其是外部调用。
- 文件系统兼容性: 非 POSIX 文件系统依赖用户配置, 文档需明确说明。

重点 PR 速览

PR 编号	标题	重要性	作者	核心变更
#2181	[3/n] Disaggregated rollout: engine-side /pull_weights	9.1	nanjiangwill	引擎侧权重同步端点, 减少 <code>slime</code> 拓扑感知
#2180	Add <code>--release-train</code>	9.0	zhuzilin	训练后释放 GPU 资源, 重建时从 checkpoint 加载
#2183	cleanup	7.7	zhuzilin	删除过时函数和参数, 增加多节点 guard
#2185	Support <code>routed_experts_start_len</code>	5.9	zhuzilin	扩展 <code>routed experts</code> 数据拼接逻辑
#2184	sync <code>source_names</code>	6.0	zhuzilin	新增数据来源追踪字段

详细说明:

- #2181: 新增 `/pull_weights` 端点, 通过调度器广播到所有 host, 应用 `delta` 并确认。
`slime` 侧移除了 `all_engine_actors` 和 `sync_local_checkpoint`。设计文档详细描述了动机和验证。

- #2180: 重构 RayTrainGroup, 延迟 Actor 初始化, 新增 release() 和 create() 方法。训练循环中根据 --release-train 释放资源, 配合 sglang HiCache 补丁释放内存。
- #2183: 清理函数包括 get_advantages_and_returns (已被 batch 版替代)、onload_weights_from_disk、parse_critic_args。移除 logprob_utils.py 中冗余参数。增加多节点环境检查, 避免错误行为。
- #2185: 在 _apply_meta_info 中读取 routed_experts_start_len, 调整行数计算, 并放宽 can_reuse_log_probs_in_loss 条件。
- #2184: 新增 get_source 函数, 在 rollout 中提取 source_names, 训练侧跳过该字段的日志计算, 更新文档。

后续建议

1. 补齐测试: 针对引擎重构、release-train、routed experts 等功能增加单元测试和集成测试, 降低回归风险。
2. 监控性能: release-train 模式涉及 Actor 组释放和重建, 需监控时间开销, 评估对训练效率的影响。
3. 文档完善: 明确非 POSIX 文件系统配置要求, 以及 source_names 在不同场景的配置方式。
4. 验证多节点: 清理 PR 中的多节点 guard 变更需在真实多节点环境验证, 确保不会引入新问题。
5. 规划后端适配: 引擎侧 /pull_weights 与 sglang 绑定, 后续其他后端 (如 Megatron) 是否采用类似模式需评估。