

PR #2589 完整报告

radixark/miles

Do not abort in-flight requests when bumping the engine weight version

合并时间: 2026-08-18 08:07

原文链接: <http://prhub.com.cn/radixark/miles/pull/2589>

执行摘要

- 一句话: 权重版本更新不再中止在途请求, 修复大规模样本丢弃
- 推荐动作: 值得精读。虽然只有一行改动, 但 PR body 提供了完整的根因链与实测数据, 能帮助理解 sglang 的 pause、abort、版本 bump 生命周期与 rollout 生产量的关系。建议后续补充针对请求 payload 的回归测试, 并关注 sglang 上游参数语义变化。

功能与动机

PR body 描述了完整的故障链: `_finalize_and_resume_engines` (mixin.py) 调用 `engine.update_weight_version(new_version)`, 而 wrapper 只传 `{"new_version": ...}`, sglang 的 `UpdateWeightVersionReqInput` 默认 `abort_all_requests=True`, handler 在 `continue_generation` 前执行 `abort_request(abort_all=True)`, 与 `in_place` 暂停模式下 "保留在途请求待新权重续跑" 的设计直接冲突。实测 16 节点 GLM-5.2 agentic 运行中, 每次更新丢约 65 个请求, 38% 样本 ABORTED, `train_wait_time` 升至 137-340s。

实现拆解

实现分为三步拆解:

1. 问题定位: `miles/backends/sglang_utils/sglang_engine.py` 的 `update_weight_version` 通过 `_make_request` 发送 payload 时只有 `new_version`, sglang 服务端 `UpdateWeightVersionReqInput` 的 `abort_all_requests` 默认值为 `True`, 于是版本 bump 后立刻执行 `abort_request(abort_all=True)`, 把 pause 阶段保留的在途请求全部杀死。此调用是在 `_finalize_and_resume_engines` (mixin.py) 中为 full-param 与 LoRA 更新统一引入的, 旧路径将 `weight_version` 放在 `update_weights_from_distributed` payload 里, 从未命中该端点。
2. 修复方式: 在 `update_weight_version` 的请求体里显式追加 `"abort_all_requests": False`, 让服务端尊重 pause 模式已决定的在途请求策略——在 `in_place` 模式下这些请求会保留下来, 换上新权重继续解码, 不再产出 `finish_reason: "abort"` 的 partial message。
3. 配套情况: 本次没有新增测试, 也没有配置或部署配套改动。建议后续补一个针对 `update_weight_version` 请求 payload 的快速断言, 防止未来重构再次落回默认 abort 行为。

关键文件:

- `miles/backends/sglang_utils/sglang_engine.py` (模块 推理引擎; 类别 `source`; 类型 `core-logic`; 符号 `update_weight_version`): 这是本次修复的唯一改动文件, 直接决定引擎权重版本 bump 时是否保留在途请求。

关键符号: `update_weight_version`

关键源码片段

`miles/backends/sglang_utils/sglang_engine.py`

这是本次修复的唯一改动文件, 直接决定引擎权重版本 bump 时是否保留在途请求。

```
# miles/backends/sglang_utils/sglang_engine.py
def update_weight_version(self, weight_version: str):
    """推进引擎侧权重版本号, 供 full-param 与 LoRA 更新共用。

    关键点: sglang 服务端的 `UpdateWeightVersionReqInput` 默认
    `abort_all_requests=True`, handler 会在 `continue_generation` 之前
    执行 `abort_request(abort_all=True)`。在 `pause_generation_mode=in_place`
    下, 暂停逻辑故意保留在途请求等新权重续跑, 因此这里必须显式关闭
    abort, 避免版本 bump 把保留的请求全部打掉。
    """
    return self._make_request(
        "update_weight_version",
        # 之前只传 new_version, sglang 会使用默认 True 中止全部在途请求
        {"new_version": weight_version, "abort_all_requests": False},
    )
```

评论区精华

本 PR 没有留下实质性 review 评论: claude[bot] 仅自动提醒手动审核配置, Zhichenzzz 与 guapisolo 均直接批准。最有价值的讨论其实集中在 PR body 中: 作者明确指出 `update_weight_version` 是 finalize 阶段新增的统一版本 bump 路径, 其默认 `abort_all=True` 与 `in_place` pause 的目标互相矛盾; 同时用 16 节点 GLM-5.2 运行数据证明 38% 样本 ABORTED 和 `rid_to_state` KeyError 均由 `abort_all` 扫描与 `_wait_one_response` 唤醒竞争引起。

- review 审核情况与 PR 背景分析 (other): Zhichenzzz 与 guapisolo 均批准, 无未解决事项。

风险与影响

- 风险: 改动位于 `miles/backends/sglang_utils/sglang_engine.py`, 属于推理引擎封装层, 影响所有走 `update_weight_version` 的权重更新路径。主要风险有三点: 1) 依赖 sglang 服务端 `UpdateWeightVersionReqInput` 参数语义, 若上游升级改变字段名或默认行为, 可能出现兼容问题; 2) 未新增测试, 缺少对请求 payload 的快速断言, 未来重构可能回归; 3) 显式关闭 abort 后, 任何依赖“版本切换即清理在途请求”的调用方行为会改变, 目前仓库内未发现此类假设, 但仍需留意。整体回归风险低, 因为改动仅在请求参数层面, 且恢复的是 July 基线行为。

- 影响：影响范围覆盖 class-based rollout + sglang 引擎的权重更新流程，尤其是 16 节点以上的 agentic 训练。修复后可保留每次权重更新时的在途解码请求，混合版本轮次重新出现，样本 ABORTED 率从 38% 回落，训练等待时间从 137–340s 大幅下降，同时消除 rid_to_state KeyError 日志。对用户而言恢复了正常的 rollout 生产量；对团队而言需要在 sglang engine 层持续关注参数语义变化。
- 风险标记：核心路径变更，缺少测试覆盖，依赖 sglang 参数语义

关联脉络

- PR #2522 Make the class-based rollout the default and convert legacy path to env var gated: 该 PR 引入 _finalize_and_resume_engines 统一调用 update_weight_version，是本次回归的来源；本 PR 是对其行为修正。