

PR #2340 完整报告

THUDM/slime

fix(agent): abort timed-out SGLang requests via router workers

合并时间: 2026-09-03 19:54

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2340>

执行摘要

- 一句话: 修复 agent 路径中通过 router 中止 SGLang 请求的失效问题。
- 推荐动作: 本 PR 修复了一个关键的资源泄露问题, 虽然变更不大, 但对 agent 功能的正确性很重要, 值得精读。特别关注 abort 逻辑的兼容性处理 (404 时直接 abort) 和错误处理。建议后续补充自动化测试以覆盖 router 和直连两种场景。

功能与动机

在 agent rollout 路径中, 每个 agent turn 通过 SGLang router 发送请求。当客户端取消或超时时, 必须显式中止对应的 SGLang 请求以释放调度器槽位和 KV-cache 资源。PR body 指出: 原清理逻辑向 router 发送 `POST <router>/abort_request`, 但 router 不支持该端点, 返回 404 且异常被静默忽略, 导致原始请求可能在 worker 上继续解码, 引发资源占用、重复计算和长尾延迟。

实现拆解

- 封装 abort 逻辑: 在 `slime/agent/adapters/common.py` 中, 将 `call_sglang_generate` 函数内原有的内联 abort 代码替换为对新增辅助函数 `_abort_sglang_request` 的调用, 保持异常处理不变, 但将 abort 逻辑抽离。
- 查询 worker 列表: 在 `_abort_sglang_request` 函数内, 首先尝试获取 `sglang_url/workers` 端点。若返回 404 (表明原 URL 直接指向 SGLang 服务), 则直接向该 URL 发送 abort 请求, 保持向后兼容。
- 并发 abort 所有 worker: 若成功获取 worker 列表, 提取每个 worker 的 URL, 并使用 `asyncio.gather` 并发向每个 worker 发送带有原始请求 rid 的 `/abort_request`, 确保只中止特定请求。
- 错误处理: 整个 abort 函数用 `try-except` 包裹, 避免 abort 过程中的异常影响主流程, 仅记录警告日志。
- 验证: 作者进行了编译检查 (`py_compile`)、格式检查 (`git diff --check`) 以及使用 fake router 和 workers 的手动异步测试, 但未添加自动化测试。

关键文件:

- `slime/agent/adapters/common.py` (模块 Agent 适配层; 类别 source; 类型 core-logic; 符号 `_abort_sglang_request`, `abort_worker`): 本次变更的唯一源码文件, 修复了 abort 逻辑的核心问题, 新增了 `_abort_sglang_request` 函数。

关键符号: `_abort_sglang_request`, `abort_worker`

关键源码片段

`slime/agent/adapters/common.py`

本次变更的唯一源码文件，修复了 `abort` 逻辑的核心问题，新增了 `_abort_sglang_request` 函数。

以下是新增的 `_abort_sglang_request` 函数完整实现，展示了如何查询 `worker` 列表并并发 `abort`:

```
async def _abort_sglang_request(sglang_url: str, rid: str, logger: logging.Logger) -> None:
    """Abort one request on a worker or every worker behind the router."""
    try:
        timeout = aiohttp.ClientTimeout(total=5)
        async with aiohttp.ClientSession(timeout=timeout) as session:

            async def abort_worker(url: str) -> None:
                """向单个 worker 发送 abort 请求，异常静默处理。"""
                try:
                    async with session.post(f"{url}/abort_request", json={"rid": rid}):
                        pass
                except Exception:
                    pass

            # 尝试获取 worker 列表
            async with session.get(f"{sglang_url}/workers") as response:
                if response.status == 404:
                    # 404 表示 sglang_url 可能直接指向 SGLang 服务，直接 abort
                    await abort_worker(sglang_url)
                    return
                response.raise_for_status()
                payload = await response.json()

            # 校验响应格式
            if not isinstance(payload, dict):
                logger.warning("Invalid SGLang workers response: url=%s rid=%s", sglang_url, rid)
                return

            # 提取所有 worker 的 URL
            worker_urls = [
                worker["url"].rstrip("/")
                for worker in payload.get("workers", [])
                if isinstance(worker, dict) and isinstance(worker.get("url"), str) and worker["url"]
            ]

            # 并发向所有 worker 发送 abort 请求
            await asyncio.gather(*(abort_worker(url) for url in worker_urls))
    except Exception as exc:
```

```
# 整个 abort 过程的异常被捕获并记录，不影响主流程
logger.warning(
    "SGLang abort request failed: url=%s rid=%s error=%s",
    sglang_url,
    rid,
    exc,
)
```

在 `call_sglang_generate` 函数中，原有的内联 abort 代码被替换为调用该函数：

```
except (asyncio.CancelledError, aiohttp.ClientError, asyncio.TimeoutError) as e:
    # free the sglang slot eagerly on client cancel/timeout, else the
    # orphaned generation keeps occupying KV until its own length cap
    logger.debug("[%s] sid=%s rid=%s turn aborted: %s", adapter.log_prefix, session_id, rid,
        type(e).__name__)
    await _abort_sglang_request(sglang_url, rid, logger)
    raise
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低，变更范围小且封装良好。主要风险点：
 1. 若 router 的 `/workers` 端点响应格式变化（如字段名变更），可能导致 worker 提取失败，但代码已做类型检查和异常处理。
 2. 并发 abort 多个 worker 可能带来轻微性能开销，但 abort 是异常路径，影响有限。
 3. 如果 worker 数量很多，可能增加 abort 延迟，但使用了异步并发，且设置了 5 秒超时。
 - 影响：影响 agent rollout 路径中的请求超时处理。修复后，可避免因 abort 失效导致的 KV-cache 资源泄露、重复计算和长尾延迟，提升系统稳定性和资源利用率。影响范围限于使用 SGLang router 的 agent 场景。
 - 风险标记：资源泄露修复，兼容性处理，缺少自动化测试

关联脉络

- PR #2272 feat: support streaming external rollouts: 该 PR 为外部 SGLang rollouts 添加了流式生成支持与请求级中止，与本 PR 同样涉及 SGLang abort 机制，但针对不同路径（外部 rollout vs agent）。