

# PR #2228 完整报告

radixark/miles

examples: make the agent-server trial timeout configurable

合并时间: 2026-08-07 05:56

原文链接: <http://prhub.com.cn/radixark/miles/pull/2228>

## 执行摘要

- 一句话: agent-server 调用超时改为可配置, 默认 7200s 修复超时倒置
- 推荐动作: 值得精读: 虽然只是 2 个文件的示例级修复, 但 PR body 对超时倒置的因果链分析 (客户端先超时 → trial 记 aborted → 沙箱与并发槽位继续占用 → 整个 GRPO group 被过滤 → 梯度信号损失) 非常清晰, 是理解 agentic rollout 跨层故障代价的好材料。值得关注的设计决策: 权威超时与兜底超时的分层、通过 `--extra-env-vars` 向 rollout actor 透传配置的通用手法。可结合 #919 与 #2233 (同目录后续重命名) 一起看示例线的演进。

## 功能与动机

PR body 明确指出: `swe_agent_function.run` 硬编码 3600s 天花板, 而 README 都建议用 `--agent-timeout 5400` 启动 agent server, 且 `httpx` 客户端 `timeout=None`, 因此 3600s 就是每个 trial 的实际上限, 任何高于 3600s 的 server 配置都不生效。该倒置的代价不只是常量名错位: 60-90 分钟的 trial 会被 trainer 记为 aborted 并返回 None; agent server 仍继续运行该 trial, 沙箱和 `--max-concurrent` 槽位最多再被占用 30 分钟, 在配额受限的沙箱后端会推高后续 trial 的创建失败率; 更严重的是 aborted-sample 动态过滤器会丢弃该 sample 所属的整个 GRPO group, 在 `--rollout-batch-size 4` 时一次迟到 trial 就移除该 step 四分之一的梯度信号。问题由 nblintao 在评审 #1919 时报告。

## 实现拆解

变更共 2 个文件、+24/-2, 单一 commit, 无 review 评论, 属于小范围示例修复。

1. 新增默认常量与读取函数: 在 `examples/swe-agent/swe_agent_function.py` 中新增 `_DEFAULT_AGENT_TRIAL_TIMEOUT_S = 7200` 和 `_agent_trial_timeout_s()`, 后者从 `AGENT_TRIAL_TIMEOUT` 环境变量读取, 未设置时回退到 7200s。
2. 替换 `run()` 中的硬编码超时: 在发起 `_post_agent_server` 调用前先读取 `trial_timeout_s = _agent_trial_timeout_s()`, 再传给 `asyncio.wait_for` 的 `timeout` 参数; 超时日志从写死的 "3600s" 改为动态值。设计上客户端超时只作为 agent server 失联时的兜底 (backstop), 正常情况下 server 的 `--agent-timeout` 应先行触发。
3. 更新 README 文档: 在 `examples/swe-agent/README.md` 中补充两级超时的顺序要求——`--agent-timeout` 是权威超时, 触发后 server 结束 trial 并释放沙箱; 客户端 `AGENT_TRIAL_TIMEOUT` (默认 7200s) 必须高于它, 否则客户端先超时会记录 aborted

且 sandbox 与并发槽位继续被占用、整个 GRPO group 被过滤。文档给出通过 launcher 通用环境变量钩子透传的用法: `--extra-env-vars 'AGENT_TRIAL_TIMEOUT=10800'`, 无需新增专用启动参数。

4. 测试配套: 无新增测试文件。作者在测试计划中验证了 black/isort 格式、AGENT\_TRIAL\_TIMEOUT 未设置 / 不同取值下的解析, 以及 `--extra-env-vars` 能正确透传到 rollout actor; 但未对真实 agent server 做端到端验证, 因为触发超时路径需要超过两小时的 trial。作者也明确说明默认值是对原硬编码值的严格放宽, 现有运行只会停止误 abort。

关键文件:

- `examples/swe-agent/swe_agent_function.py` (模块 智能体; 类别 source; 类型 core-logic; 符号 `_agent_trial_timeout_s`, `_DEFAULT_AGENT_TRIAL_TIMEOUT_S`, `run`) : 核心变更文件: 将 `run()` 中对 `agent-server /run` 调用硬编码的 3600s 超时替换为 AGENT\_TRIAL\_TIMEOUT 环境变量 (默认 7200s), 并新增 `_DEFAULT_AGENT_TRIAL_TIMEOUT_S` 与 `_agent_trial_timeout_s()` 两个符号。这是修复超时倒置、避免 trial 被误 abort 的关键逻辑所在。
- `examples/swe-agent/README.md` (模块 示例文档; 类别 docs; 类型 documentation) : 文档配套: 补充 `--agent-timeout` 与 AGENT\_TRIAL\_TIMEOUT 的层级顺序要求, 并给出 `--extra-env-vars` 透传示例, 修正了原文档 "keep generous" 与硬编码 3600s 之间的矛盾。

关键符号: `_agent_trial_timeout_s`, `run`

## 关键源码片段

### `examples/swe-agent/swe_agent_function.py`

核心变更文件: 将 `run()` 中对 `agent-server /run` 调用硬编码的 3600s 超时替换为 AGENT\_TRIAL\_TIMEOUT 环境变量 (默认 7200s), 并新增 `_DEFAULT_AGENT_TRIAL_TIMEOUT_S` 与 `_agent_trial_timeout_s()` 两个符号。这是修复超时倒置、避免 trial 被误 abort 的关键逻辑所在。

```
import asyncio
import logging
import os
```

```
logger = logging.getLogger(__name__)
```

```
# 客户端超时只是兜底 (backstop): 正常情况下 agent server 自己的
# --agent-timeout 会先触发, 由 server 主动结束 trial 并释放 sandbox。
# 所以默认值 7200s 必须高于示例中 agent server 的 5400s 配置,
# 否则任何高于客户端上限的 --agent-timeout 都会成为死配置: 客户端先超时,
# 把正常 trial 误判为 aborted, 并拖垮整个 GRPO group。
_DEFAULT_AGENT_TRIAL_TIMEOUT_S = 7200
```

```
def _agent_trial_timeout_s() -> int:
```

```
    """每个 trial 调用 agent server /run 的超时上限, 可通过 AGENT_TRIAL_TIMEOUT
```

环境变量覆盖。

未设置时回退到 7200s；若设置为非数字，int() 会抛 ValueError，让调用方尽早暴露配置错误。

```
"""
```

```
return int(os.environ.get("AGENT_TRIAL_TIMEOUT", _DEFAULT_AGENT_TRIAL_TIMEOUT_S))
```

```
async def run(base_url: str, prompt, request_kwargs=None, metadata=None):
```

```
... # 构造 agent_server_url / request 等前置逻辑省略
```

```
# 在 try 块之前读取本次调用的超时值
```

```
trial_timeout_s = _agent_trial_timeout_s()
```

```
try:
```

```
    response = await asyncio.wait_for(
        _post_agent_server(f"{agent_server_url}/run", request),
        timeout=trial_timeout_s,
    )
```

```
except asyncio.TimeoutError:
```

```
    # 超时后 trainer 会把该 sample 记为 aborted, aborted-sample 过滤器
```

```
    # 会丢弃它所属的整个 GRPO group；因此客户端超时必须高于 server 端
```

```
    # --agent-timeout, 让 server 先自行清理并释放沙箱。
```

```
    logger.error(f"Agent server call timed out after {trial_timeout_s}s")
```

```
    return None
```

```
except asyncio.CancelledError:
```

```
    logger.warning("Agent server call cancelled (sibling task failure?)")
```

```
    return None
```

```
except Exception as e:
```

```
    logger.error(f"Agent server call failed: {e}")
```

```
    return None
```

```
return {
```

```
    "reward": response.get("reward", 0.0),
```

```
    "exit_status": response.get("exit_status", ""),
```

```
    "eval_report": response.get("eval_report", {}),
```

```
    "agent_metrics": response.get("agent_metrics", {}),
```

```
}
```

## 评论区精华

该 PR 没有评论线程，唯一 review 来自 nblintao (APPROVED, 无正文)。核心问题由 nblintao 在评审 #1919 时发现并报告到 PR body 中：

```
"swe_agent_function.run hardcoded a 3600 s ceiling on the agent-server /run call, while both READMEs start the agent server with --agent-timeout 5400 ... any --agent-timeout above 3600 was dead configuration." "At --rollout-batch-size 4 one late trial removes a quarter of the step's gradient signal."
```

PR body 还明确了设计决策：server 自己的 `--agent-timeout` 应该先触发并自行清理 trial/ 沙箱，客户端超时只是针对 server 失联的兜底，因此默认值 7200s 必须高于示例的 5400s。这个层级关系此前被 README 中 "keep `--agent-timeout` generous" 的建议直接矛盾，本次一并修正。

- 客户端 3600s 硬编码与 server `--agent-timeout` 5400 的倒置 (correctness): 客户端超时改为 `AGENT_TRIAL_TIMEOUT` 环境变量，默认 7200s 高于 5400s; README 明确 `--agent-timeout` 为权威超时、客户端为兜底，并通过 `--extra-env-vars` 透传。

## 风险与影响

- 风险：
  1. 环境变量非法值：AGENT\_TRIAL\_TIMEOUT 在 `_agent_trial_timeout_s()` 中通过 `int()` 转换且读取位置在 `try` 块之外，若用户误设为非数字（如 `abc`），`ValueError` 会向上抛出而不是被 `run()` 的 `except Exception` 捕获返回 `None`，可能导致 rollout 调用直接失败。
  2. 默认等待时间放宽：客户端超时从 3600s 放宽到 7200s，若 agent server 失联，客户端最长等待增加一小时，拖尾 trial 对训练进度的阻塞和沙箱 / 并发槽位占用时间变长；不过正常路径由 server 5400s 先行触发，实际影响可控。
  3. 无自动化测试覆盖：超时路径依赖真实 agent server 和超过两小时的 trial，CI 无法覆盖；本次变更也没有新增任何测试文件。
  4. 影响范围有限：变更只在 `examples` 目录，不触及 `miles` 核心训练 / rollout 代码。
- 影响：
  - 用户侧：使用 `swe-agent` 示例做 agentic RL 训练的用户不再因客户端 3600s 硬编码而误 abort 60-90 分钟的正常 trial，GRPO group 不再被整组误过滤，训练梯度信号质量提升；沙箱配额释放也更符合预期。
  - 系统侧：客户端超时上限放宽到 2 小时，最坏情况下 rollout actor 等待时间变长，但正常路径由 server 侧 `--agent-timeout` (5400s) 主导，两级超时的职责划分更清晰。
  - 团队侧：README 明确了 "server 权威超时 + client 兜底超时" 的层级要求，并通过 `--extra-env-vars` 展示了一种免新增启动参数的通用配置透传模式，避免后续示例再次引入同类 dead configuration。
  - 风险标记：示例目录变更，无自动化测试覆盖，AGENT\_TRIAL\_TIMEOUT 非数字时 `int()` 异常位于 `try` 之外，默认超时放宽至 7200s，失联时最坏等待变长

## 关联脉络

- PR #1919 (stacked 基础 PR, 标题未提供) : PR body 明确指出本 PR stacked on #1919 (base 分支 `shi/swe-agent-daytona-example`)，且超时倒置问题正是 nblintao 在评审 #1919 时发现并报告的。
- PR #2233 examples: rename swe-agent to swe-agent-harbor-docker: 同属 `examples/swe-agent` 示例线：#2228 修改的 `swe_agent_function.py` 与 README 所在目录，在 #2233 中被重命名为 `swe-agent-harbor-docker`，两者共同构成 `agent-server` 示例的功能完善与整理演进。