

PR #1956 完整报告

THUDM/slime

Add slime/agent/ and move sandbox impl inside

合并时间: 2026-05-27 11:56

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1956>

执行摘要

- 一句话: 将 sandbox 从示例迁移至核心 slime/agent 模块, 定义协议接口
- 推荐动作: 值得精读, 尤其关注 Sandbox 协议的设计方式 (Protocol + runtime_checkable)、环境变量优先级的处理策略以及 RPC 重试的指数退避实现。这些设计模式可复用于其他需要后端抽象的模块。

功能与动机

之前的 coding_agent_rl 示例内联了完整的 E2B sandbox 实现, 其他 agent 场景无法复用。通过将通用抽象提升到 slime/agent 层, 后续 agent 示例无需重复实现 sandbox 生命周期与 RPC 重试逻辑, 仅需实现 Sandbox 协议即可接入不同后端 (Docker、Modal 等)。

实现拆解

1. 创建核心包: 新增 slime/agent/ 包及 __init__.py, 新建 slime/agent/sandbox.py 定义 Sandbox 异步协议接口 (exec、write_file、read_file、__aenter__、__aexit__) 和 E2BSandbox 实现类。
2. 抽取通用实现: 从 examples/coding_agent_rl/sandbox.py 中移出 E2B 客户端通信、元数据解析、RPC 重试、生命期管理等通用代码至 slime/agent/sandbox.py, 保留 SWE 特定的安装和评估函数。
3. 改造示例 sandbox: examples/coding_agent_rl/sandbox.py 改为从 slime.agent.sandbox 导入 E2BSandbox 和 Sandbox, 删除原来内联的 _parse_sandbox_metadata、_is_transient_rpc_error、_rpc_retry、E2BSandbox 等, 只保留 install_node22、install_claude_code、ensure_agent_user、apply_before_repo_set_cmd、apply_pre_commands 等函数。
4. 调整生成入口: examples/coding_agent_rl/generate.py 导入从 sandbox.E2BSandbox 改为 slime.agent.sandbox.E2BSandbox, 并修复 write_text 调用为 write_file 以匹配协议接口。
5. 配套更新: 更新 README.md 说明 sandbox 位置变化和环境变量新名称; requirements.txt 将 e2b 依赖提升为公共依赖。

关键文件:

- slime/agent/sandbox.py (模块 核心层; 类别 source; 类型 core-logic; 符号 Sandbox, E2BSandbox, aenter, aexit) : 新增核心模块, 定义 Sandbox 协议和 E2BSandbox 实现

，封装 E2B 通信、元数据解析、RPC 重试等通用逻辑。

- `examples/coding_agent_rl/sandbox.py`（模块 示例层；类别 `source`；类型 `dependency-wiring`；符号 `install_node22`, `install_claude_code`, `ensure_agent_user`, `apply_before_repo_set_cmd`）：大规模重构，删除通用 E2B 实现，仅保留 SWE 业务逻辑，改为依赖 `slime.agent.sandbox`。
- `examples/coding_agent_rl/generate.py`（模块 示例层；类别 `source`；类型 `dependency-wiring`）：导入路径调整和 `write_text` 调用修复，确保与新的 `Sandbox` 协议接口一致。
- `slime/agent/__init__.py`（模块 核心层；类别 `source`；类型 `entrypoint`）：新增包初始化文件，标记 `agent` 模块。

关键符号：`Sandbox.aenter`, `Sandbox.aexit`, `Sandbox.exec`, `Sandbox.write_file`, `Sandbox.read_file`, `E2BSandbox.init`, `E2BSandbox._metadata_from_env`, `E2BSandbox._is_transient_rpc_error`, `E2BSandbox._rpc_retry`, `_getenv`, `install_node22`, `install_claude_code`, `ensure_agent_user`, `apply_before_repo_set_cmd`, `apply_pre_commands`

关键源码片段

`examples/coding_agent_rl/sandbox.py`

大规模重构，删除通用 E2B 实现，仅保留 SWE 业务逻辑，改为依赖 `slime.agent.sandbox`。

```
"""Coding-agent sandbox helpers.
```

```
The provider-agnostic sandbox contract and E2B backend live in
``slime.agent.sandbox``. This module keeps the coding-agent/SWE-specific
bootstrap, Claude Code runner, diff capture, and fresh-sandbox evaluator.
"""
```

```
from __future__ import annotations
```

```
import asyncio
```

```
import json
```

```
import logging
```

```
import lzma
```

```
import os
```

```
import shlex
```

```
import shutil
```

```
import tempfile
```

```
import time
```

```
from pathlib import Path
```

```
# 从核心层导入通用 sandbox 抽象
```

```
from slime.agent.sandbox import E2BSandbox, Sandbox
```

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

```

# 沙箱内路径（避免与镜像自带路径冲突）
_PATCH = "/workspace/__cagent_patch__.diff"
_PRE = "/workspace/__cagent_pre__.sh"
_SWEPRO_DIR = "/workspace/swepro_eval"

async def install_node22(sb: Sandbox, host_tarball: Path) -> None:
    """在沙箱内安装 Node 22（基镜像通常只有 Node 16）。


若主机上的压缩包是 .xz 格式，先解压为 .tar 缓存到临时目录，  
避免沙箱内没有 xz-utils。


    """
    host_tarball = Path(host_tarball)
    if host_tarball.suffix == ".xz":
        plain = (
            Path(tempfile.gettempdir())
            / f"coding_agent_rl.{host_tarball.stem}.tar"
        )
        if not plain.exists():
            tmp = plain.with_suffix(".tar.partial")
            with lzma.open(host_tarball, "rb") as src, open(tmp, "wb") as dst:
                shutil.copyfileobj(src, dst)
            os.replace(tmp, plain)
        host_tarball = plain
    await sb.write_file("/tmp/node22.tar", host_tarball)
    await sb.exec(
        "set -e && mkdir -p /opt/node22 && "
        "tar xf /tmp/node22.tar -C /opt/node22 --strip-components=1 && "
        "ln -sf /opt/node22/bin/node /usr/local/bin/node && "
        "ln -sf /opt/node22/bin/npm /usr/local/bin/npm && "
        "ln -sf /opt/node22/bin/npx /usr/local/bin/npx && "
        "hash -r 2>/dev/null || true && node --version && npm --version",
        user="root",
        timeout=180,
        check=True,
    )

```

评论区精华

当前 PR 无 review 评论，但关键设计决策体现在源码注释中：

- 环境变量命名：类变量同时枚举 SLIME_AGENT_SANDBOX_* 和 SWE_*, 优先使用前者，实现向后兼容。
- RPC 重试策略：固定退避基数为 1.0 秒，总等待预算 3 秒，注释明确解释不开放为可配的理由（避免用户误调导致更大窗口或过长 stall）。
- Sandbox 协议选择 `runtime_checkable` Protocol 而非 ABC，降低实现门槛。
- 暂无高价值评论线程

风险与影响

- 风险:

1. 向后兼容风险: 仍支持旧 `SWE_*` 环境变量, 但需确认所有原有使用处均被正确映射。
`write_text` 改为 `write_file` 可能破坏外部依赖该旧方法的自定义脚本。
2. 测试覆盖不足: 本次变更未增加针对 `slime/agent/sandbox.py` 的单元测试, 核心 RPC 重试逻辑和元数据解析缺少回归保护。
3. 接口稳定期: Sandbox 协议刚提取至公共空间, 后续可能因新需求调整接口, 引入不稳定风险。
 - 影响: 对 `coding_agent_rl` 示例用户: 需要更新导入路径和环境变量 (实际向下兼容, 旧环境变量仍工作)。对系统: 无直接运行时影响。对团队: 为后续 agent 场景 (如 `SWE-bench`、`Agent-QA`) 提供了可复用的 `sandbox` 抽象, 减少重复编码和维护成本。
 - 风险标记: 向后兼容: 环境变量和接口对齐需验证, 测试覆盖不足: 无新增测试, 公共模块接口稳定期

关联脉络

- PR #1923 [examples] add `coding_agent_rl`: agent-in-sandbox RL minimal demo: 首次引入 `coding_agent_rl` 示例, 包含原始内联 `E2BSandbox` 实现, 本 PR 从中抽取通用部分。
- PR #1958 Move `coding_agent_rl`'s helper function to `sandbox.py`: 先前将 `generate.py` 中的辅助函数移至 `sandbox.py`, 为本 PR 的提取奠定了基础。
- PR #1957 Minor refactor for `coding agent rl` logic and remove `SWE_LIST_TRAJECTORY`: 同一功能线上的重构, 调整 agent RL 生成逻辑和配置, 与本 PR 共同完善 `coding_agent_rl` 示例。