

PR #1896 完整报告

radixark/miles

Add a shell launch script test harness for future protection

合并时间: 2026-08-09 18:21

原文链接: <http://prhub.com.cn/radixark/miles/pull/1896>

执行摘要

- 一句话: 为 shell 启动脚本新增可安全重放的测试 harness
- 推荐动作: 值得精读。虽然全部是测试代码, 但 harness 的设计决策很有学习价值: 用 PATH shim 拦截真实命令、按 fork 顺序保证记录确定性、轮询 /proc 过滤僵尸进程、冻结环境并拒绝解冻、路径占位符替换。建议与 #1899 (shell 脚本外部命令快照) 和 #1901 (Python 启动脚本 harness) 连读, 能看到同一思路向两个方向的扩展。对需要维护 launch 脚本的工程师, 理解 fake_bin 拦截与占位符机制是必要的。

功能与动机

PR body 仅标注 "Part of #1837"。结合 #1837 的清单, 该 PR 服务于“为未来提供保护”: shell 启动脚本直接执行 ray、pkill、torchrun、hf download 等命令, 任何后续重构 (如 #1897 修复、#1909 参数展开) 都可能在不经意间改变行为, 而此前 CI 无法安全运行这些脚本。harness 的设计目标就是让真实脚本在完全不触碰真实系统 (不下载权重、不启动训练、不杀死进程) 的前提下跑完, 并记录所有外部命令调用。

实现拆解

1. 沙箱入口与目录布局: sh_harness.py 定义 run_launch_script(script, sandbox, extra_env, timeout) 作为统一入口, 在 sandbox 下建立 fake_bin (shim 目录)、capture (命令调用记录文件)、workdir (脚本工作目录) 三个关键位置。
2. 命令 shim 拦截: _write_shims() 为 22 个危险/外部命令 (apt、curl、docker、git、hf、pkill、ray、torchrun、wget 等) 生成 bash shim, 将 argv 以控制符分隔追加到 capture 后退出 0; 对 python/python3 输出 1000000、date 输出固定时间, 使脚本中的 GPU 等待轮询立即满足, 避免测试挂起。
3. 环境冻结: _FROZEN_ENV 固定 HOME、LANG、MASTER_ADDR、NODE_RANK、WANDB_KEY 等; PATH 前置 fake_bin; _reject_unfreezing() 断言调用方传入的 extra_env 不得覆盖冻结变量或 capture 通道, 防止测试环境泄露导致快照漂移。
4. 执行与记录顺序: 以 start_new_session=True 启动 bash, 超时后 os.killpg 清理整个进程组; _wait_until_the_script_leaves_nothing_running() 轮询 /proc 进程组并过滤 Z 状态僵尸进程, 处理 & 后台命令与命令替换孤儿进程; _parse_capture() 按 pid (fork 顺序) 而非追加顺序排序记录, _sanitize() 将 sandbox 与 REPO_ROOT 替换为占位符。

5. 结果模型与测试覆盖：LaunchScriptRun 提供 invocations_of 与 ray_job_submit_argv 查询 API，format_invocations() 生成快照文本；test_sh_harness.py 以真实脚本 scripts/run-qwen3-4B.sh 与合成脚本验证破坏性命令拦截、后台记录顺序（连续 20 次）、环境冻结断言、路径占位与重放确定性；__init__.py 使 tests/fast/launch_scripts 成为可导入测试包。

关键文件：

- tests/fast/launch_scripts/sh_harness.py（模块 测试沙箱；类别 test；类型 test-coverage；符号 LaunchScriptRun, invocations_of, ray_job_submit_argv, run_launch_script）：整个 PR 的核心：提供 run_launch_script 沙箱执行与命令 shim 记录机制，是后续快照测试与脚本重构的安全网。
- tests/fast/launch_scripts/test_sh_harness.py（模块 沙箱测试；类别 test；类型 test-coverage；符号 TestRunLaunchScriptOnARealScript, TestRunLaunchScriptOnABackgroundingScript, TestRunLaunchScriptEnvironmentFreeze, TestRunLaunchScriptOnTheShimEdgeCases）：harness 的自测套件，覆盖真实脚本与合成脚本的边界（后台命令、命令替换、GPU 等待循环、环境冻结、路径占位），验证 harness 的确定性与安全性。
- tests/fast/launch_scripts/__init__.py（模块 测试包；类别 test；类型 test-coverage）：将 launch_scripts 提升为可导入测试包，供 harness 与用例之间模块化引用。

关键符号：run_launch_script, _write_shims, _parse_capture, _sanitize, _reject_unfreezing, _wait_until_the_script_leaves_nothing_running, _live_pids_of_group, format_invocations, LaunchScriptRun.invocations_of, LaunchScriptRun.ray_job_submit_argv

关键源码片段

tests/fast/launch_scripts/test_sh_harness.py

harness 的自测套件，覆盖真实脚本与合成脚本的边界（后台命令、命令替换、GPU 等待循环、环境冻结、路径占位），验证 harness 的确定性与安全性。

```
import pytest
```

```
from tests.fast.launch_scripts.sh_harness import run_launch_script
```

```
class TestRunLaunchScriptOnABackgroundingScript:
```

```
    @pytest.fixture
```

```
    def script(self, tmp_path):
```

```
        # 一个会后台启动 sglang server 的合成脚本，用于验证 shim 对 `&` 子进程的记录
```

```
        script = tmp_path / "backgrounding.sh"
```

```
        script.write_text(
```

```
            ""#!/bin/bash
```

```
set -ex
```

```
python3 -m sglang.launch_server --port 13141 >/dev/null 2>&1 &
```

```
curl -sf http://127.0.0.1:13141/health_generate
```

```

ray job submit --address="http://127.0.0.1:8265" -- python3 train.py
"""

    )
    return script

def test_a_backgrounded_command_is_still_recorded(self, script, tmp_path):
    # bash 退出时不回收 `&` 子进程，过早读 capture 会漏掉它，这里验证必须记录到
    run = run_launch_script(script, sandbox=tmp_path / "sandbox", timeout=30)

    assert run.returncode == 0
    assert run.invocations_of("python3")[0][1:3] == ["-m", "sglang.launch_server"]

@pytest.mark.parametrize("attempt", range(20))
def test_backgrounding_does_not_perturb_the_recorded_order(self, script, tmp_path, attempt)
:
    # 快照断言精确序列，`&` 不得打乱记录的先后顺序，连续 20 次验证稳定性
    run = run_launch_script(script, sandbox=tmp_path / f"sandbox-{attempt}", timeout=30)

    assert [argv[0] for argv in run.invocations] == ["python3", "curl", "ray"]

def test_a_command_backgrounded_inside_a_substitution_does_not_hang_the_run(self, tmp_
path):
    # 命令替换中后台的命令会被孤儿化，若 PID 1 不回收则以僵尸形式残留，killpg 仍可见；
    # harness 必须能识别这种情况并正常结束而不是一直等待
    script = tmp_path / "substitution.sh"
    script.write_text(
        """#!/bin/bash

set -ex
start_server() {
    python3 -m sglang.launch_server --port "$1" >/dev/null 2>&1 &
    echo "/tmp/server-$1.log"
}
LOG=$(start_server 13141)
curl -sf http://127.0.0.1:13141/health_generate
ray job submit --address="http://127.0.0.1:8265" -- python3 train.py "$LOG"
"""
    )

    run = run_launch_script(script, sandbox=tmp_path / "sandbox", timeout=30)

    assert run.returncode == 0
    assert [argv[0] for argv in run.invocations] == ["python3", "curl", "ray"]

```

评论区精华

本 PR 没有产生实质性 review 讨论：审核人 yueming-yuan 直接批准（APPROVED），评论区内无提问或争议。harness 中的关键设计权衡——按 fork 顺序而非追加顺序记录、跳过僵尸进程、拒绝覆盖冻结环境——并非讨论产物，而是由作者在代码注释与测试用例中固化（例如

test_sh_harness.py 中专门测试 & 后台命令不扰乱记录顺序的用例）。因此本节的洞察主要来自代码证据而非对话。

- 暂无高价值评论线程

风险与影响

- 风险:

1. sh_harness.py 的 _live_pids_of_group() 直接遍历 /proc，是 Linux 专属实现，在 macOS 等非 Linux 开发环境下 fast 测试会失败或行为异常。
2. _SHIMMED_COMMANDS 覆盖有限：脚本若调用未拦截命令（如 sudo、ssh、kill）且带副作用，harness 可能误放行真实系统操作。
3. _parse_capture() 按 pid 排序假设 fork 顺序稳定，对复杂脚本中多线程同时 fork 的场景可能不成立。
4. _sanitize() 只替换 sandbox 与 REPO_ROOT，若脚本输出时间戳或其他绝对路径，快照仍会漂移。
5. test_sh_harness.py 对 run-qwen3-4B.sh 的具体断言 (--num-layers 36、/root/Qwen3-4B) 与真实脚本强耦合，脚本演进时需同步更新。
6. 整体仅新增测试文件，无生产路径回归风险。- 影响：对用户无运行时影响（纯测试资产）。对团队而言，shell 启动脚本的修改将进入快照保护范围：后续任何改动导致命令序列、环境变量或路径变化时，测试会立即失败，合并成本上升但回归风险下降。对 CI 系统，fast 测试套件新增 shell 脚本执行测试，且依赖 Linux 环境的 /proc 机制。该 PR 是 #1837 系列中“启动脚本可测试化”的起点，直接影响 #1897、#1898、#1899、#1901 等后续 PR 的实施方式。- 风险标记：测试基础设施首次引入，依赖 Linux /proc，快照与真实脚本强耦合，shim 列表需随脚本演进

关联脉络

- PR #1899 Snapshot the external commands of every shell launch script: 该 PR 的直接消费者：用 sh_harness 对每个 shell 启动脚本建立外部命令快照，本 PR 为其提供安全执行与记录基础。
- PR #1901 Snapshot the commands and generated configs of every python launch script: 同系列测试基建：将相同思路扩展到 Python 启动脚本（py_harness），与本 PR 的 sh_harness 互补。
- PR #1898 Derive the miles checkout location instead of hardcoding it in launch scripts: 同属 #1837 的启动脚本重构线，与本 PR 一样服务于脚本可复现性与快照化。
- PR #1897 Fix various launch scripts errors about missing line concatenations or paths: 同一批次对 launch 脚本的修复，harness 可防止此类错误再次引入。