

PR #1906 完整报告

radixark/miles

Snapshot the launchers that build their own command line

合并时间: 2026-08-09 18:46

原文链接: <http://prhub.com.cn/radixark/miles/pull/1906>

执行摘要

- 一句话: 为自构建命令的启动器新增快照测试
- 推荐动作: 值得精读。重点看 `install_shell_recorder` 的冻结策略 (PID、sleep、makedirs) 和 `iter_self_executing_launchers` 的“按行为发现”设计, 它把‘哪些启动器需要保护’变成了可断言的测试。对负责启动脚本重构的工程师, 这套快照是正确的第一道防线; 后续 PR #1907-#1911 都建立在此基础上。

功能与动机

本 PR 是跟踪 issue #1837 “refactoring and enhancements”中的一环 (PR body 注明 Part of #1837)。在整理启动脚本时发现, `p2p_weight_transfer/run.py`、`formal_math/run_minimal.py` 这类启动器不经过 `command_utils`, 而是自己构建完整命令行并直接调用 `subprocess` 或 `shell` 脚本, 既有快照测试 (针对 `shell` 启动器和 `python` 启动器) 无法覆盖。测试 `docstring` 明确指出 “These launchers build their whole command line by hand, so only a snapshot pins it”, 即只有快照能把这类手写命令固定下来, 防止后续重构悄悄改变训练启动命令。

实现拆解

1. 扩展测试基架 (`tests/fast/launch_scripts/py_harness.py`): 新增 `iter_self_executing_launchers`, 以“不属于 `scripts/run_*.py` 约定且源码文本包含 `ray job submit`”的行为特征发现自执行启动器; 新增 `install_shell_recorder`, 通过 `monkeypatch subprocess.run`、`time.sleep`、`os.makedirs`、`os.getpid/os.getppid` 录制命令并冻结不确定性来源, 使录制结果可复现。
2. 新增快照测试 (`tests/fast/launch_scripts/test_self_executing_launchers.py`): 定义 `LauncherCase` 与参数化 `fixture recorded`, 为 `p2p_weight_transfer` 的 11 个模型 `profile x p2p/broadcast` 模式以及 `formal_math` 的 `import` 模式共 23 个 case 执行启动器入口并把命令录制结果与快照比较; 另含重跑一致性、最终必须 `ray job submit`、发现覆盖三个维度断言。
3. 生成 23 个快照文件 (`tests/snapshots/launch_scripts/self_executing/`): 按“启动器路径 / 场景名 .txt”的结构存放完整录制 (含环境清理、`ray` 启动、`ray job submit` 携带的完整训练参数), 作为 CI 比对基线; 这些文件同时充当手写命令的“文档”。
4. 配套约束: `TestDiscovery` 保证“按行为发现”与“已覆盖 case”完全一致, 新增手写启动器会立刻被测试点名; 同时显式声明无法被沙箱化的 `cmd_prepare` 入口点并校验其硬编码

/root/models 路径，避免测试盲区被掩盖。

关键文件：

- tests/fast/launch_scripts/test_self_executing_launchers.py (模块 启动器；类别 test；类型 test-coverage；符号 LauncherCase, recorded, TestEverySelfExecutingLauncher, test_commands_match_snapshot)：本 PR 的核心：为自执行启动器新增快照测试与发现断言，是 23 个快照的消费方。
- tests/fast/launch_scripts/py_harness.py (模块 测试基架；类别 test；类型 test-coverage；符号 iter_self_executing_launchers, install_shell_recorder, fake_run)：新增录制器与发现函数，是快照测试的基架。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/GLM-5/p2p.txt (模块 快照；类别 docs；类型 documentation)：代表性快照，固定了 GLM-5 在 p2p 传输模式下的完整训练命令，同类快照共 23 个。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/GLM-5/broadcast.txt (模块 快照；类别 docs；类型 documentation)：与 p2p 对应的 broadcast 模式快照，体现同一启动器不同模式的差异。

关键符号：iter_self_executing_launchers, install_shell_recorder, fake_run, recorded, test_commands_match_snapshot, test_reruns_produce_identical_recordings, test_the_launcher_submits_a_ray_job, test_every_self_executing_launcher_has_at_least_one_case, test_the_uncovered_entrypoint_is_named_and_still_uncoverable

关键源码片段

tests/fast/launch_scripts/py_harness.py

新增录制器与发现函数，是快照测试的基架。

自执行启动器：不满足 scripts/run_*.py 约定，但源码里包含 ray job submit 的 Python 文件

```
def iter_self_executing_launchers() -> list[Path]:
    """Launchers that reach the shell themselves rather than through command_utils."""
    roots = [REPO_ROOT / root for root in ("scripts", "examples", "tools")]
    convention = {script.path for script in iter_py_launch_scripts()}
    return sorted(
        path
        for root in roots
        for path in root.rglob("*.py")
        if path not in convention and "ray job submit" in path.read_text(errors="replace")
    )
```

这类启动器持有自己的 subprocess 句柄，不会经过 command_utils 的录制器，因此单独拦截 subprocess.run

```
def install_shell_recorder(monkeypatch, sandbox: Path) -> Recording:
    """A launcher holding its own subprocess handle never touches the recorded command_utils helpers."""
    recording = Recording(commands=[], pseudo_files=[])
```

```
# fake_run 记录命令并返回一个足够大的 GPU 数量，让启动器里的等待循环立即通过
def fake_run(command, *args, **kwargs):
    recording.commands.append(command if isinstance(command, str) else " ".join(command))
    return subprocess.CompletedProcess(
        args=command, returncode=0, stdout=_GPU_COUNT_ANY_WAIT_LOOP_ACCEPTS,
        stderr=""
    )

monkeypatch.setenv("MILES_LOG_DIR", str(sandbox))
monkeypatch.setattr(subprocess, "run", fake_run)
monkeypatch.setattr(time, "sleep", lambda seconds: None)
monkeypatch.setattr(os, "makedirs", lambda path, **kwargs: None)
monkeypatch.setattr(os, "getpid", lambda: _FROZEN_PID)
monkeypatch.setattr(os, "getppid", lambda: _FROZEN_PPID)

return recording
```

评论区精华

本 PR 没有人审 review 评论（review_comments_count 为 0），GitHub 上仅有一条 Gemini Code Assist 的停用提示，不构成技术讨论。真正有价值的“讨论”藏在测试代码的 docstring 里，例如“这些启动器完全手写命令行，只有快照能把它钉住”“启动器会把自身 PID 嵌进清理命令，所以快照只有在冻结 PID 后才会稳定”，这些是测试设计约束的自我说明。

- 暂无高价值评论线程

风险与影响

- 风险：快照与模型脚本强耦合：23 个快照记录了 p2p_weight_transfer 针对 11 个模型 profile 的完整参数，任何参数默认值、模型脚本路径或配置 key 变化都会让 CI 失败，维护成本集中在 tests/snapshots/launch_scripts/self_executing/ 下。录制有盲区：install_shell_recorder 只拦截 subprocess.run，若启动器改用 os.system、subprocess.Popen 等路径则录不到；fake_run 固定返回 GPU 数量，依赖真实命令输出的启动器可能被“假阳性”通过。发现机制依赖文本匹配：iter_self_executing_launchers 靠源码里出现 ray job submit 字符串判断，注释或文档字符串误出现可能造成误报 / 漏报，目前依赖 TestDiscovery 兜底。此外 cmd_prepare 因硬编码 /root/models 无法沙箱化，该入口点没有被录制覆盖。
- 影响：开发者修改任何自执行启动器（尤其是 p2p_weight_transfer 与 formal_math 示例）时必须同步更新对应快照，否则 CI 会失败；CI 新增 23 个快照断言，运行时间略有增加。对团队而言，这套基线把长尾的手写启动命令显式化，使 #1837 系列后续重构（如 #1907 修复 rotary_base、#1910 用 Python 替换 shell 配置）有明确的回归网。对生产系统无影响（纯测试变更）。
- 风险标记：快照维护成本高，环境敏感测试，文本发现机制，monkeypatch 依赖，存在无法覆盖的入口点

关联脉络

- PR #1907 Fix p2p profile's rotary_base not reaching the model script it configures: 同一 p2p_weight_transfer 启动器与快照目录，后续修复了本 PR 快照暴露出的参数传递缺陷。
- PR #1908 Snapshot test the argv of all model scripts: 同一快照测试系列的下一环，将快照覆盖扩展到所有模型脚本 argv。
- PR #1909 Expand the model args in python before building the command: 同一命令构建链路的重构，依赖本 PR 的快照基线验证行为不变。
- PR #1910 Replace the model config shell scripts with python: 同一系列中将 shell 模型配置改造成 Python，本 PR 的快照提供回归保护。
- PR #1911 Quote the model args miles inlines into the launch command: 同一系列中修复内联参数引用问题，同样依赖快照验证。
- PR #2279 Run the launch script snapshot tests by hand instead of in CI: 后续把启动脚本快照测试移出 CI，改为手动执行，直接关联本 PR 新增的测试。