

PR #1908 完整报告

radixark/miles

Snapshot test the argv of all model scripts

合并时间: 2026-08-09 18:47

原文链接: <http://prhub.com.cn/radixark/miles/pull/1908>

执行摘要

- 一句话: 为全部模型脚本 argv 增加快照测试
- 推荐动作: 值得精读。重点阅读 model_args_harness.py 中如何用最小化环境执行 shell 展开并抽取 argv, 以及 test_model_args.py 中 TestDiscovery 的设计——用「集合一致性」防止快照文件失联的思路值得借鉴。此外, 这组快照测试是理解后续 #1909、#1910、#1911 一系列重构的必读前提。

功能与动机

PR body 明确说明这是 #1837 (重构与增强跟踪 issue) 的一部分。提交信息进一步解释了动机: 「The next ops rewrite all 62 scripts/models/*.sh into python. Once the shell versions are gone there is no source of truth left to prove the rewrite was faithful, so record the argv each of them expands to now.」也就是说, 在 shell 脚本被删除之前必须先把当前展开结果固化为 golden 文件, 否则重写后没有任何证据能证明新 Python 实现与旧 shell 行为一致。此外, 这批快照还让 18 个此前没有任何 launcher 快照覆盖的模型获得了第一层基础覆盖。

实现拆解

1. 新增 argv 提取工具: 在 tests/fast/launch_scripts/model_args_harness.py 中实现 iter_model_types() (通过 glob 扫描 scripts/models/*.sh 得到模型类型列表) 和 expand_model_args() (以最小化环境变量执行 source script && printf 展开 MODEL_ARGS 数组, 逐行输出)。最小化环境 (PATH/HOME/LANG/LC_ALL) 保证快照结果不受宿主机环境干扰。
2. 新增参数化快照测试: tests/fast/launch_scripts/test_model_args.py 中 TestEveryModelType.test_model_args_match_snapshot 对每个模型类型执行展开并与 tests/snapshots/model_args/{model_type}.txt 比较; test_model_args_are_flags_and_values 校验首 token 以 -- 开头且所有 token 无空白, 防止 argv 拼接后隐式分裂。
3. 新增发现一致性测试: TestDiscovery.test_every_model_is_discovered_and_snapshotted 断言 iter_model_types() 返回的模型集合与快照目录中的 .txt 文件集合完全一致, 并断言模型数量 > 60, 防止新模型脚本或快照文件被静默遗漏。
4. 生成 60+ 个 golden 快照文件: 为 deepseek-v4-pro、deepseek-v4-flash、deepseek-v32、moonlight、glm4.7-flash、glm5-744B-A40B、kimi-k2 等所有模型脚本逐一生成

tests/snapshots/model_args/*.txt, 内容为 JSON 序列化后的逐 token argv。

5. 配套后期演进：本 PR 的快照文件在后续 #1909、#1910、#1911 中被持续重新生成（如为 run.py 命令增加 --model_args 内联 base64 参数等），承担了「重构是否 faithful」的验证职责。

关键文件：

- tests/fast/launch_scripts/test_model_args.py（模块 模型参数测试；类别 test；类型 test-coverage；符号 TestEveryModelType, test_model_args_match_snapshot, test_model_args_are_flags_and_values, TestDiscovery）：核心测试文件：通过参数化测试将每个模型脚本的展开 argv 与 golden 文件比对，并用 TestDiscovery 保证模型发现与快照文件集合一致，防止重构后模型参数漂移。
- tests/fast/launch_scripts/model_args_harness.py（模块 模型参数工具；类别 test；类型 test-coverage；符号 iter_model_types, expand_model_args）：测试基础设施：定义了 iter_model_types 与 expand_model_args，是快照数据来源，后续 Python 化重写必须复现其输出。
- tests/snapshots/model_args/deepseek-v4-pro.txt（模块 快照文件；类别 docs；类型 documentation）：代表 60+ 个 golden 快照文件，记录了 deepseek-v4-pro 的完整 argv，是后续 Python 化重写必须逐字节复现的基准。

关键符号：iter_model_types, expand_model_args, test_model_args_match_snapshot, test_model_args_are_flags_and_values, test_every_model_is_discovered_and_snapshotted

关键源码片段

tests/fast/launch_scripts/test_model_args.py

核心测试文件：通过参数化测试将每个模型脚本的展开 argv 与 golden 文件比对，并用 TestDiscovery 保证模型发现与快照文件集合一致，防止重构后模型参数漂移。

```
import json
```

```
import pytest
```

```
from tests.fast.launch_scripts.model_args_harness import expand_model_args, iter_model_types
from tests.fast.launch_scripts.sh_harness import REPO_ROOT, assert_matches_snapshot
```

```
# 快照目录集中放在 tests/snapshots/model_args 下，便于统一管理
_SNAPSHOT_DIR = REPO_ROOT / "tests" / "snapshots" / "model_args"
```

```
# 模块加载时一次性发现全部模型脚本，保证参数化用例集合固定
_MODEL_TYPES = iter_model_types()
```

```
class TestEveryModelType:
```

```
    @pytest.mark.parametrize("model_type", _MODEL_TYPES)
    def test_model_args_match_snapshot(self, model_type: str) -> None:
```

```

"""The golden argv of every model, so a later rewrite of the model definitions cannot drift.
"""

# 逐 token JSON 序列化后按行拼接, 保证快照可读且能精确还原 shell 展开结果
actual = "\n".join(json.dumps(token) for token in expand_model_args(model_type)) + "\n"

assert_matches_snapshot(_SNAPSHOT_DIR / f"{model_type}.txt", actual, model_type)

@pytest.mark.parametrize("model_type", _MODEL_TYPES)
def test_model_args_are_flags_and_values(self, model_type: str) -> None:
    """Consumers split the args on whitespace, so a token that contains any would silently
    become two."""
    tokens = expand_model_args(model_type)

    # 调用方按空白切分参数, 因此任何含空白的 token 都会静默分裂, 这里直接封死该可能
    assert tokens
    assert tokens[0].startswith("--")
    assert all(token == token.strip() and " " not in token for token in tokens)

class TestDiscovery:
    def test_every_model_is_discovered_and_snapshotted(self) -> None:
        """A model that stops matching the discovery glob would otherwise lose its golden file
        silently."""
        # 若模型脚本改名或新增, 而快照文件未同步, 此断言会立即失败, 避免 golden 文件失联
        snapshotted = {path.stem for path in _SNAPSHOT_DIR.glob("*.txt")}

        assert set(_MODEL_TYPES) == snapshotted
        assert len(_MODEL_TYPES) > 60

```

tests/fast/launch_scripts/model_args_harness.py

测试基础设施: 定义了 `iter_model_types` 与 `expand_model_args`, 是快照数据来源, 后续 Python 化重写必须复现其输出。

```

import subprocess

from tests.fast.launch_scripts.sh_harness import REPO_ROOT

# 模型脚本统一放在 scripts/models 目录下
MODEL_SCRIPT_DIR = REPO_ROOT / "scripts" / "models"

# 刻意剔除 LOGNAME/USER 等环境变量, 避免宿主环境干扰快照展开结果
_ENV_WITHOUT_THE_MODEL_ARGS_KNOBS = {
    "PATH": "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
    "HOME": "/root",
    "LANG": "C",
    "LC_ALL": "C",
}

def iter_model_types() -> list[str]:

```

```
# 以文件名（不含 .sh）作为模型类型标识，例如 deepseek-v4-pro
return sorted(path.stem for path in MODEL_SCRIPT_DIR.glob("*.sh"))
```

```
def expand_model_args(model_type: str) -> list[str]:
    """The golden files are taken from this shell expansion; whatever replaces it must reproduce
    them."""
    # source 脚本后按行打印 MODEL_ARGS 数组，得到展开后的 argv 逐 token 列表
    script = MODEL_SCRIPT_DIR / f"{model_type}.sh"
    result = subprocess.run(
        f'source "{script}" && printf "%s\\n" "${{MODEL_ARGS[@]}}"',
        shell=True,
        executable="/bin/bash",
        env=_ENV_WITHOUT_THE_MODEL_ARGS_KNOBS,
        capture_output=True,
        text=True,
        check=True,
    )
    return result.stdout.splitlines()
```

评论区精华

该 PR 没有实质性的 review 讨论。仅有 [gemini-code-assist\[bot\]](#) 的一条系统通知，说明 Gemini Code Assist 的消费者版本已停用，所有代码审查活动正式终止，因此没有来自人类的评论或争议。值得关注的是 PR 作者同时也是合并者，单人完成了整个 op 链的推进，并依赖快照测试本身的失败结果来驱动后续调整。

- 无人工 review 讨论 (other): 唯一评论是机器人通知，无实际技术讨论，PR 由作者自行合并。

风险与影响

- 风险：
 1. 快照更新纪律风险：60+ 个 golden 文件完全依赖「手动重新生成」来保持与脚本一致，若后续修改模型脚本而忘记更新快照，CI 会失败，这是设计使然，但也意味着每次模型参数调整都会产生大量快照文件变更，可能带来 review 噪音。
 2. 对宿主环境的敏感性：expand_model_args() 通过 subprocess.run 执行 shell 脚本，虽然设置了最小化环境，但如果本地开发机的 bash 版本或环境变量处理与 CI 不一致，可能出现假阳性失败。快照文件本身与 shell 展开行为强绑定，后续 Python 化重写必须逐字节复现这些输出。
 3. 文件数量膨胀风险：tests/snapshots/model_args/ 下新增 60+ 个文本文件，加上后续 #1909 对 shell 快照的补充，仓库内快照文件总数显著增加，需要留心快照文件的管理策略（如是否纳入 pre-commit 自动重生）。
 4. 数量断言脆弱性：assert len(_MODEL_TYPES) > 60 引入了对模型数量的硬编码下限，若未来清理或合并模型脚本导致总数降至 60 以下，测试会失败，但这种情况实际发生概率很低，风险可控。- 影响：对用户与开发者的影响：该 PR 不改变任何运行时行为，所有变更都在 tests/ 下，因此对线上训练、rollout 等功能无直接影响。对团队的影响：建

立了模型脚本重构的「行为基准线」，任何对 scripts/models/*.sh 的修改都会在 CI 中被快照测试自动校验，能显著降低「改坏模型参数而无人察觉」的风险。对后续重构的影响：tests/snapshots/model_args/*.txt 成为 #1909、#1910、#1911 等 PR 的验收依据，实际推动了 launcher 快照、模型配置 Python 化等系列重构的落地。- 风险标记：快照文件更新依赖手动重生成，测试对 bash 环境敏感，快照文件数量大幅膨胀，数量断言硬编码下限

关联脉络

- PR #1909 Expand the model args in python before building the command: 紧随本 PR 的下一步重构：在 Python 中预展开模型参数，命令构建不再依赖 shell 展开，本 PR 的快照文件为这一改动提供了回归基准。
- PR #1910 Replace the model config shell scripts with python: 将 scripts/models/*.sh 重写为 Python，必须通过本 PR 的快照测试证明重写后 argv 与 shell 时代完全一致。
- PR #1911 Quote the model args miles inlines into the launch command: 本 PR 的快照文件在 #1911 中被重新生成（如 deepseek-v4 系列快照反映内联 base64 参数），体现了快照作为重构验证工具的持续使用。
- PR #1899 Snapshot the external commands of every shell launch script: 同属 #1837 重构链，为 shell 启动脚本建立外部命令快照，与本 PR 的模型 argv 快照互为补充，共同构筑启动链路行为的完整基线。
- PR #2279 Run the launch script snapshot tests by hand instead of in CI: 后续将部分启动脚本快照测试移出 CI 改为手动执行，与本 PR 的快照测试策略形成连续性演进。