

PR #1898 完整报告

radixark/miles

Derive the miles checkout location instead of hardcoding it in launch scripts

合并时间: 2026-08-09 18:24

原文链接: <http://prhub.com.cn/radixark/miles/pull/1898>

执行摘要

- 一句话: 启动脚本不再硬编码 checkout 路径, 改为运行时推导
- 推荐动作: 值得精读, 尤其是 `tests/fast/launch_scripts/test_shell_script_hygiene.py` 的测试设计: 用不到 20 行代码就固化了一个跨脚本的工程约定, 且刻意避开对 git 的依赖, 思路清晰。对于需要维护大量 shell 启动脚本的团队, 这是一个可复制的“脚本卫生检查”样板。另外可以留意本 PR 与 #1911 的引号化改动之间的一致性, 理解脚本健壮性的两个侧面: 路径可推导 + 参数可引用。

功能与动机

关联 Issue #1837 列出的重构目标之一是消除启动脚本中不可复现、依赖固定容器环境的假设; 新增测试的 docstring 直接点明动机: “A script that assumes one absolute checkout only runs inside one container image.” 即脚本一旦被复制到别的容器或目录, 硬编码的 `/root/miles` 或 `/workspace/miles` 就会失效。提交信息也体现了意图: 用“Find the shell scripts without shelling out to git”的方式扫描脚本, 避免对 git 环境的额外依赖。

实现拆解

本 PR 的变更可分为 5 步:

1. 新增卫生测试作为回归防线: 在 `tests/fast/launch_scripts/test_shell_script_hygiene.py` 中新增 `test_no_shell_script_hardcodes_the_checkout_location`, 用 `REPO_ROOT.rglob("*.sh")` 扫描 `scripts` 与 `examples` 两个目录下的全部 shell 脚本, 凡是文本中出现 `/root/miles` 或 `/workspace/miles` 的脚本都被视为违规, 并断言违规列表为空。测试刻意不调用 `git ls-files`, 避免对 git 环境的依赖。
2. 统一推导 miles 仓库根目录: 将 `examples/experimental/strands_sglang/strands_qwen3_8b.sh` 等脚本中的 `source "/root/miles/scripts/models/..."` 改为先计算 `SCRIPT_DIR`, 再基于脚本所在目录的相对层级推导出 `MILES_ROOT`, 例如 `$(cd -- "$(dirname -- "${BASH_SOURCE[0]}")/../../.." && /dev/null && pwd)`, 进而 `source "${SCRIPT_DIR}/../../scripts/models/qwen3-8B.sh"`。不同目录深度的脚本使用了不同的相对路径层数, 并保留 `${MILES_DIR:-..}` 这种环境变量覆盖机制。
3. 调整 PYTHONPATH 等运行时环境变量: 以 `strands_qwen3_8b.sh` 为代表, ray 的 `RUNTIME_ENV_JSON` 中 `PYTHONPATH` 原本写死 `/root/miles`, 改为引用推导出的 `${MILES_ROOT}`, 保证 ray job 提交到集群节点时路径仍然有效。

4. 为 train.py 调用加引号：qwen3_5_35b_selfdistill 系列脚本中 python3 `${MILES_DIR}/train.py` 改为 python3 "`${MILES_DIR}/train.py`"，防止推导出的路径含空格或特殊字符时被 shell 拆分，这与后续 #1911 对 model args 的引号化是同一方向。
5. 配套变更：本次还顺带修复了若干脚本中路径无法解析的问题（合并自前序提交），并更新了 examples/infra_features/p2p_weight_transfer/run-glm4.5-air-8node-profile.sh 等 8 个 P2P profile 脚本中的 MILES_ROOT 赋值。没有新增配置或部署配套。

关键文件：

- tests/fast/launch_scripts/test_shell_script_hygiene.py（模块 脚本卫生；类别 test；类型 test-coverage；符号 test_no_shell_script_hardcodes_the_checkout_location）：本次变更的回归防线：新增扫描脚本，断言 scripts 与 examples 下所有 *.sh 不包含硬编码 checkout 路径，是该 PR 唯一的新增测试文件，也是后续同类问题的守护者。
- examples/experimental/strands_sglang/strands_qwen3_8b.sh（模块 实验示例；类别 other；类型 core-logic）：改动最多的示例脚本：既把 source 模型配置改为相对路径，又把 ray 环境变量里的 PYTHONPATH 从 /root/miles 改为推导的 MILES_ROOT，完整展示了路径推导的两个使用场景。
- examples/on_policy_distillation/qwen3_5_35b_selfdistill/phase1_rlvr_teacher.sh（模块 蒸馏示例；类别 other；类型 core-logic）：展示了 MILES_DIR 默认值由 /root/miles 改为 BASH_SOURCE 推导，并将 python3 `${MILES_DIR}/train.py` 加引号，是路径推导与引号化结合的典型代表。
- examples/infra_features/p2p_weight_transfer/run-glm4.5-air-8node-profile.sh（模块 权重迁移；类别 other；类型 core-logic）：8 个 P2P profile 脚本的代表：MILES_ROOT 从 /root/miles 改为基于脚本位置的推导，随后 source 仓库内模型配置，保证 profile 脚本可以移植到任意 checkout。
- examples/on_policy_distillation/run-qwen3-8B-opd.sh（模块 蒸馏示例；类别 other；类型 core-logic）：三个 run-qwen3-8B-opd 系列脚本的代表：source "`/root/miles/scripts/models/qwen3-8B.sh`" 改为 SCRIPT_DIR 相对路径，是本次批量替换最典型的场景。

关键符号：test_no_shell_script_hardcodes_the_checkout_location

关键源码片段

tests/fast/launch_scripts/test_shell_script_hygiene.py

本次变更的回归防线：新增扫描脚本，断言 scripts 与 examples 下所有 *.sh 不包含硬编码 checkout 路径，是该 PR 唯一的新增测试文件，也是后续同类问题的守护者。

```
# tests/fast/launch_scripts/test_shell_script_hygiene.py
# 从 sh_harness 复用仓库根目录定位；path 为 pathlib.Path 实例，由 harness 约定提供
from tests.fast.launch_scripts.sh_harness import REPO_ROOT

# 历史硬编码的 checkout 绝对路径；只要脚本中再度出现，本测试就直接失败
_HARDCODED_CHECKOUTS = ("/root/miles", "/workspace/miles")

def test_no_shell_script_hardcodes_the_checkout_location():
```

```

""" 只在一个绝对 checkout 位置才能运行的脚本，换容器镜像后就失效。"""
# 扫描 scripts 与 examples 下所有 *.sh，找出文本中出现硬编码路径的文件
offenders = [
    path.relative_to(REPO_ROOT).as_posix()
    for root in (REPO_ROOT / "scripts", REPO_ROOT / "examples")
    for path in root.rglob("*.sh")
    for text in [path.read_text(errors="replace")]
    if any(hardcoded in text for hardcoded in _HARDCODED_CHECKOUTS)
]

# 有任何一个违规脚本，列表非空，断言失败并列出违规文件
assert offenders == []

```

examples/experimental/strands_sglang/strands_qwen3_8b.sh

改动最多的示例脚本：既把 source 模型配置改为相对路径，又把 ray 环境变量里的 PYTHONPATH 从 /root/miles 改为推导的 MILES_ROOT，完整展示了路径推导的两个使用场景。

```

# examples/experimental/strands_sglang/strands_qwen3_8b.sh
# 由脚本自身位置推导出仓库根目录，不再硬编码 /root/miles
SCRIPT_DIR="$(cd -- "$(dirname -- "${BASH_SOURCE[0]}")" &>/dev/null && pwd)"
MILES_ROOT="$(cd -- "$(dirname -- "${BASH_SOURCE[0]}")/../../" &>/dev/null && pwd)"

# source 仓库内的模型配置脚本，改用相对路径，保证任意 checkout 位置都可运行
source "${SCRIPT_DIR}/../../scripts/models/qwen3-8B.sh"

# 构建 ray 运行时环境时，PYTHONPATH 内的 miles 路径也改为推导值，
# 这样 ray job 落在集群节点上时能找到同一个代码根目录
RUNTIME_ENV_JSON="{ \"env_vars\": {
    \"PYTHONPATH\": \"/${MILES_ROOT}/${SCRIPT_DIR}/${MILES_ROOT}\",
    \"CUDA_DEVICE_MAX_CONNECTIONS\": \"1\",
    \"NCCL_NVLS_ENABLE\": \"${HAS_NVLINK}\"
  }}"

```

评论区精华

评审过程本身没有产生技术争论：yueming-yuan 直接 APPROVED，没有留下评论；仓库内仅有的评论来自 [gemini-code-assist\[bot\]](#)，声明其代码审查服务已停用，与本 PR 技术内容无关。

有信息量的讨论实际沉淀在提交信息里，可以视为作者的设计意图：

提交信息明确写出“Find the shell scripts without shelling out to git”——意即不使用 `git ls-files` 来枚举脚本，而是直接用 `rglob("*.sh")` 从文件系统扫描，从而让测试在非 git 工作区也能运行。

另一个值得注意的点是提交序列把“修复路径”“加引号”和“推导路径”拆成多个 squash 提交（#1895、#1896、#1897 的相关内容先并入），说明作者将这组改动视为一个互相咬合的链条，最终合并为本 PR 时保持了单一职责。

- 评审无实质讨论，直接 APPROVED (other): 评审通过，未留下技术争议或待办事项。

风险与影响

- 风险：本 PR 的核心风险来自 shell 路径推导的脆弱性：
 1. 依赖 BASH_SOURCE 与脚本层级：MILES_ROOT/MILES_DIR 的推导基于脚本所在目录与仓库根目录的相对深度（../..、../... 等）。一旦将来移动这些示例脚本到别的目录层级，推导路径会静默指向错误位置；现有测试只检查“没有硬编码路径”，并不验证推导出的路径是否真的指向仓库根，属于覆盖盲区。
 2. ray 集群跨节点问题：脚本在本机推导出的 MILES_ROOT 会写入 RUNTIME_ENV_JSON 的 PYTHONPATH，但 ray 集群其他节点不一定在相同路径挂载代码。虽然保留了 \${MILES_DIR:-...} 环境变量覆盖，但如果用户直接使用默认推导值，跨节点时仍可能失败。
 3. 硬编码检测列表不完整：测试只拦截 /root/miles 和 /workspace/miles 两个已知字符串，未来若出现 /opt/miles、~/miles 等新写法，测试无法拦截。
 4. 兼容性：部分脚本原本默认 MILES_DIR=/workspace/miles，现在默认值变为推导结果，依赖该目录内容的存量环境在未设置 MILES_DIR 时行为会变化，需通过环境变量显式覆盖。
 - 影响：影响范围集中在示例与工具启动脚本：examples/experimental/strands_sglang、examples/on_policy_distillation（含 qwen3_5_35b_selfdistill 系列）、examples/infra_features/p2p_weight_transfer 的多个 profile 脚本、examples/lora、scripts 下的单个脚本，共 19 个文件。

对用户而言，最大的收益是脚本不再绑定固定容器镜像路径，可以从任意 checkout 位置直接运行，降低了复制脚本到新环境时的隐性故障；需要付出的成本是脚本对自身目录结构变化的敏感度提高。对团队而言，新增的卫生测试把“禁止硬编码 checkout 路径”变成一条可执行的约定，后续新增脚本若违规会在 CI 中立即暴露，为同批次的快照测试（#1899、#1901、#1906）提供了干净的基线。由于这些都是示例和工具脚本，不涉及训练核心路径，整体影响可控。

- 风险标记：依赖脚本相对层级位置，硬编码检测列表有限，跨节点路径依赖环境变量覆盖，新增测试未验证推导路径正确性

关联脉络

- PR #1899 Snapshot the external commands of every shell launch script: 同属 #1837 op-chain，且共享 REPO_ROOT 与 sh_harness 基础设施，本 PR 清理硬编码路径为快照测试提供干净基线。
- PR #1901 Snapshot the commands and generated configs of every python launch script: 同一批次快照测试系列，本 PR 消除的硬编码路径正是这些快照可复现的前提。
- PR #1897 Fix various launch scripts errors about missing line concatenations or paths: 同批次前序改动，修复了脚本路径解析问题，部分提交被 squash 进本 PR 的 head 分支，两者相互依赖。
- PR #1911 Quote the model args miles inlines into the launch command: 与本 PR 共同推动启动命令可复现性：本 PR 为推导出的 train.py 路径加引号，#1911 为内联 model args 加引号，方向一致。