

PR #1905 完整报告

radixark/miles

Remove non-reproducible file arguments by supporting inline base64 payloads

合并时间: 2026-08-09 18:46

原文链接: <http://prhub.com.cn/radixark/miles/pull/1905>

执行摘要

- 一句话: 以 base64 内联配置替代临时文件, 增强启动命令可复现性
- 推荐动作: 建议精读。本 PR 展示了如何在不引入共享文件系统的前提下让启动命令可复现, 并配套了完整的失败路径测试 (损坏 base64 必须报错而非静默解出空配置)。值得关注的设计是 PSEUDO_FILE_PREFIX 前缀约定与 resolve_file_arg 单一入口, 以及 encode_pseudo_file 替代临时文件写入的确定性策略; 后续参数若需要文件与内联两种形态都可以复用这一模式。

功能与动机

本 PR 是 #1837 中“op-chain”的一部分, 目标是让 miles 的启动命令可复现。旧实现 save_to_temp_file 生成 /tmp/miles_temp_file__.yaml, 命令内容依赖时间和随机数, 快照测试无法稳定对比; 同时 launcher 与 worker 之间必须共享文件系统才能读到配置。PR 提交说明直接指出“Remove non-reproducible file arguments”, 通过内联 base64 载荷使命令内容与文件系统状态解耦。

实现拆解

1. 新增统一解析入口 miles/utils/file_arg_utils.py, 定义 PSEUDO_FILE_PREFIX = "base64:" 与 resolve_file_arg(value); 调用方用同一函数即可同时消费普通路径与内联载荷。
2. 在 miles/utils/external_utils/command_utils.py 中删除 save_to_temp_file (依赖 time.time() 与 random.randrange() 生成不可复现路径), 新增确定性 encode_pseudo_file(text), 把内容编码为 base64: 前缀字符串; 同时移除不再需要的 import time。
3. 接入参数解析层 miles/utils/arguments.py: --eval-config 与 --custom-config-path 的消费点 (_resolve_eval_datasets、miles_validate_args) 改用 resolve_file_arg, help 文本同步提示内联载荷写法。
4. 更新各 launch script (scripts/run_mcore_fsdp.py、scripts/run_deepseek_v32.py、scripts/run_joy_ai_llm_flash.py、scripts/run_qwen3_30b_a3b.py、scripts/run_qwen3_4b.py、scripts/amd/run_qwen3_30b_a3b.py、scripts/run_deepseek_v4.py、scripts/run_glm45_355b_a32b.py 等), 将 eval/tis/custom config 的传递从 save_to_temp_file 改为 encode_pseudo_file。

5. 更新 `examples/experimental/eval/eval_delegate_rollout.py`: `_get_delegate_client` 对内联载荷直接解析, `_build_delegate_client` 改为接收文本并用 `OmegaConf.create` 构建; 内联载荷不参与 `mtime` 缓存。
6. 配套测试: 新增 `tests/fast/utils/test_file_arg_utils.py` (覆盖普通路径、内联解码、多行 UTF-8、locale 无关性、缺失路径报错、损坏载荷报错); `tests/fast/utils/test_command_utils.py` 将 `TestSaveToTempFile` 替换为 `TestEncodePseudoFile` (round-trip、确定性、命令行往返); 新增 `tests/fast/examples/experimental/eval/test_eval_delegate_rollout.py` (内联与普通路径均可达 `delegate`); `tests/fast/launch_scripts/py_harness.py` 的 `fake` 由 `fake_save_to_temp_file` 改为 `fake_encode_pseudo_file`, 并同步更新快照。

关键文件:

- `miles/utils/file_arg_utils.py` (模块 参数解析; 类别 `source`; 类型 `core-logic`; 符号 `resolve_file_arg`): 新增的统一参数解析入口, 定义 `base64`: 前缀协议与 `resolve_file_arg` 核心逻辑, 是本次变更的基础。
- `miles/utils/external_utils/command_utils.py` (模块 命令工具; 类别 `source`; 类型 `core-logic`; 符号 `encode_pseudo_file`, `save_to_temp_file`): 用确定性 `encode_pseudo_file` 替换不可复现的 `save_to_temp_file`, 是启动命令可复现的关键改动点。
- `miles/utils/arguments.py` (模块 参数层; 类别 `source`; 类型 `dependency-wiring`): 参数解析层从此统一经过 `resolve_file_arg`, 保证主解析器与下游消费方对同一参数行为一致。
- `examples/experimental/eval/eval_delegate_rollout.py` (模块 评估代理; 类别 `source`; 类型 `core-logic`; 符号 `_build_delegate_client`, `_get_delegate_client`): `eval delegate` 是内联载荷的第一个下游消费者, 修改后普通路径与内联配置都能正确构建 `delegate client`。
- `tests/fast/utils/test_file_arg_utils.py` (模块 参数测试; 类别 `test`; 类型 `test-coverage`; 符号 `TestResolveFileArg`, `test_reads_a_plain_file_path`, `test_decodes_an_inline_base64_payload`, `test_round_trips_multiline_utf8_content`): 核心工具函数的关键路径测试, 包含损坏载荷必须报错等防御性测试, 保障协议边界。

关键符号: `resolve_file_arg`, `encode_pseudo_file`, `_get_delegate_client`, `_build_delegate_client`

关键源码片段

`miles/utils/file_arg_utils.py`

新增的统一参数解析入口, 定义 `base64`: 前缀协议与 `resolve_file_arg` 核心逻辑, 是本次变更的基础。

```
# miles/utils/file_arg_utils.py
import base64
from pathlib import Path
```

```
# 前缀约定: 凡是 base64: 开头的参数都被视为内联配置, 而不是文件路径
PSEUDO_FILE_PREFIX = "base64:"
```

```
def resolve_file_arg(value: str) -> str:
```

"""读取命令行参数：普通文件路径或 base64：内联载荷。

内联载荷让启动命令自带配置文本，不依赖 launcher 与 worker 共享文件系统，从而保证命令内容可复现，快照测试不会因随机 /tmp 路径而抖动。

"""

```
if value.startswith(PSEUDO_FILE_PREFIX):
    # validate=True 让非法 base64 直接抛 binascii.Error,
    # 避免默认行为静默丢弃非法字符、解出空配置
    return base64.b64decode(value[len(PSEUDO_FILE_PREFIX) :], validate=True).decode()
# 普通路径统一按 UTF-8 读取，与 OmegaConf 的历史行为保持一致
return Path(value).read_text(encoding="utf-8")
```

miles/utils/arguments.py

参数解析层从此统一经过 resolve_file_arg，保证主解析器与下游消费方对同一参数行为一致。

```
# miles/utils/arguments.py 中 --custom-config-path 的消费点
if args.custom_config_path:
    # 普通路径与 base64：内联载荷统一通过 resolve_file_arg 取得内容
    data = yaml.safe_load(resolve_file_arg(args.custom_config_path)) or {}
    for k, v in data.items():
        if hasattr(args, k):
            logger.info(
                f"Warning: Argument {k} is already set to {getattr(args, k)}, will override with {v}."
            )
```

评论区精华

本 PR 没有实质性的代码 review 讨论：reviewer yueming-yuan 直接批准，未留下评论；gemini-code-assist 仅在关联 issue 中说明其审查功能已终止。因此没有可提炼的技术交锋，设计决策（base64 前缀、失效必须报错等）主要通过测试注释传达。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 命令行长度膨胀：base64 编码比原文约膨胀 33%，对较长配置（如 --te-precision-config-file 或 --custom-config-path）可能逼近 ARG_MAX，多节点 ray submit 场景尤需注意；涉及 scripts/run_deepseek_v32.py、scripts/run_mcore_fsdp.py 等。
 2. 编码与 locale：resolve_file_arg 对普通文件路径统一按 UTF-8 read_text；若用户配置文件为 GBK 等其他编码，将抛 UnicodeDecodeError，与旧 OmegaConf.load 行为可能不一致（虽然测试断言 OmegaConf 也按 UTF-8 读）。
 3. 内联载荷绕过 mtime 缓存：eval_delegate_rollout._get_delegate_client 对内联载荷直接构建 client，不缓存；同一进程内多次调用且配置不变时可能重复创建 delegate client，带来额外资源开销。

4. 敏感信息暴露：配置内容直接出现在进程命令行中，可能被 ps、shell 历史记录捕获，涉及安全审查。
5. 快照与测试同步风险：py_harness 的 fake 与各脚本的调用点必须保持一致，若后续有人继续引用已删除的 save_to_temp_file 会立即失败。 - 影响：影响范围中等偏广：涉及 22 个文件，覆盖所有主要 launch script、参数解析层、eval delegate 示例和快照测试。用户侧不再需要 launcher 与 worker 共享文件系统，小配置可直接内联传递，启动命令更稳定、可复现；但配置内容会出现在命令行中，敏感信息需评估。团队侧获得统一的 resolve_file_arg 入口，后续新增“文件或内联”二选一的参数只需复用该工具，测试也更易做确定性快照。 - 风险标记：命令行长度膨胀，内联载荷绕过缓存，配置暴露于命令行，大范围脚本接线替换

关联脉络

- PR #1906 Snapshot the launchers that build their own command line: 属于同一 op-chain，且共享启动命令快照测试基础设施；本 PR 的 py_harness fake 改动与 #1906 的自构建启动器快照直接相关。
- PR #1908 Snapshot test the argv of all model scripts: 快照测试同一批模型脚本 argv，本 PR 将临时文件替换为内联 base64 后，模型脚本的快照随之更新，两者互为依赖。
- PR #1909 Expand the model args in python before building the command: 同样聚焦命令构建的确定性与可复现性，是 op-chain 中相邻步骤。
- PR #1911 Quote the model args miles inlines into the launch command: 后续接续本 PR 的内联参数思路，为内联的 model args 增加 shell 引号处理，修正参数被 glob 展开的问题。
- PR #2279 Run the launch script snapshot tests by hand instead of in CI: 快照测试生命周期后续调整，反映本 PR 所建快照机制的演进。