

PR #1907 完整报告

radixark/miles

Fix p2p profile's rotary_base not reaching the model script it configures

合并时间: 2026-08-09 18:47

原文链接: <http://prhub.com.cn/radixark/miles/pull/1907>

执行摘要

- 一句话: 修复 P2P profile 的 rotary_base 未到达模型脚本的问题
- 推荐动作: 值得精读。这个 PR 展示了一个最小但完整的“配置到达真实命令行”修复闭环: 把单点配置抽成两个纯函数, 用真实 bash 展开而非 mock 来测试, 并以快照防回归。对编写启动脚本、命令构造工具或示例类代码的工程师有直接借鉴意义。重点阅读 build_model_args_command 中“先设变量再 source”的构造方式, 以及测试中“测试真实命令而不是逻辑副本”的策略。

功能与动机

该 PR 是追踪 Issue #1837 「refactoring and enhancements」下的第 13 项子任务。PR 标题直接点出问题: p2p profile 的 rotary_base 没有到达它配置的模型脚本。测试注释也明确说明了根因: "The knob has to survive into the shell that expands MODEL_ARGS, not only into ray's runtime env"——即模型脚本在 source 时读取环境变量来决定 --rotary-base, 而此前 rotary_base 只存在于 ray runtime env 中, shell 展开 MODEL_ARGS 时根本看不到它, 导致实际训练参数与 profile 声明不一致 (例如 Qwen3-235B-A22B-Instruct-2507 声明的 5000000 可能静默丢失)。

实现拆解

第 1 步: 定位根因

在 `examples/infra_features/p2p_weight_transfer/run.py` 的 `cmd_run` 中, `cfg.rotary_base` 此前只被写入 ray job 的 runtime env (`env_vars["MODEL_ARGS_ROTARY_BASE"]`), 而真正携带模型参数的 shell 命令是 `source "scripts/models/{model_type}.sh" && ray job submit ... ${MODEL_ARGS[@]}`。模型脚本在 source 时才能决定 `--rotary-base` 的值, 但该环境变量在 source 那一刻并不存在于 shell 中, 于是实际参数与 profile 声明脱节。

第 2 步: 抽出统一构造逻辑

新增两个函数, 让两处注入点共用同一数据源:

- `build_model_args_env(cfg) -> dict[str, str]`: 把 `cfg.rotary_base` 映射为 `{"MODEL_ARGS_ROTARY_BASE": str(value)}`, None 时返回空字典;
- `build_model_args_command(cfg) -> str`: 用 `shlex.quote` 把键值对拼成 `VAR=value` 前缀, 再接 `source ".../scripts/models/{model_type}.sh"`, 保证变量在 source 时就位。

`cmd_run` 中对应修改：`runtime env` 改用 `env_vars.update(model_args_env)`，提交命令改用 `build_model_args_command(cfg)`；同时把 `import shlex` 从提交分支内提升到模块顶层，删除原先重复的内联逻辑。

第 3 步：新增单元测试

新增 `tests/fast/examples/infra_features/p2p_weight_transfer/test_run.py`（含 3 个空 `__init__.py` 包保证 `pytest` 可发现）：

- 用 `importlib` 直接加载 `run.py` 为模块，避免复制逻辑；
- `expand_model_args` 辅助函数把 `build_model_args_command` 的返回值交给真实 `/bin/bash` 执行，再 `printf` 打印 `"${MODEL_ARGS[@]}"`，测试的是 `run.py` 真实命令而非逻辑副本；
- 用例覆盖：不固定 `rotary_base` 的 `profile` 不注入任何覆盖（Qwen3-4B）、固定值的 `profile` 生成正确的 `MODEL_ARGS_ROTARY_BASE`（Qwen3-235B-A22B-Instruct-2507）、参数化验证 3 个固定值的 `profile` 展开出的 `--rotary-base` 与声明一致、以及所有 `profile` 都能展开成非空且以 `--` 开头的 `argv`（防止模型脚本改名后提交无效作业）。

第 4 步：更新快照

`tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/` 下 6 个快照（GLM-4.5-Air、Qwen3-235B-A22B-Instruct-2507、Qwen3-30B-A3B 各自的 `broadcast/p2p` 模式）在命令行中新增 `MODEL_ARGS_ROTARY_BASE=<值>` 前缀，并同步更新 `runtime env JSON`，保持快照测试绿色。

第 5 步：合并验证

14 个 `commit` 中 12 个属于 #1837 系列其余子任务（`typo` 修复、`shell harness`、快照基建、`base64 payload` 等），最后一个 `merge commit` 与 `main` 对齐，最终由 `yueming-yuan` 批准合并。

关键文件：

- `examples/infra_features/p2p_weight_transfer/run.py`（模块 示例运行；类别 `source`；类型 `core-logic`；符号 `build_model_args_command`, `build_model_args_env`）：核心修复文件：新增 `build_model_args_env` / `build_model_args_command`，把 `rotary_base` 注入 `source` 前的 `shell` 环境，解决 `profile` 参数无法到达模型脚本的问题。
- `tests/fast/examples/infra_features/p2p_weight_transfer/test_run.py`（模块 示例测试；类别 `test`；类型 `test-coverage`；符号 `run_module`, `expand_model_args`, `test_model_args_env_is_empty_when_the_profile_pins_no_rotary_base`, `test_model_args_env_carries_the_rotary_base_a_profile_pins`）：新增测试文件：通过真实 `bash` 执行 `run.py` 构造的命令并展开 `MODEL_ARGS`，验证 `rotary_base` 确实到达模型脚本，防止回归。
- `tests/fast/examples/__init__.py`（模块 测试结构；类别 `test`；类型 `test-coverage`）：新增空包文件，使 `tests/fast/examples` 下的测试目录可被 `pytest` 发现。

- tests/fast/examples/infra_features/__init__.py (模块 测试结构; 类别 test; 类型 test-coverage) : 新增空包文件, 支持多层目录下的测试发现。
- tests/fast/examples/infra_features/p2p_weight_transfer/__init__.py (模块 测试结构; 类别 test; 类型 test-coverage) : 新增空包文件, 支持该测试目录被 pytest 收集。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/GLM-4.5-Air/broadcast.txt (模块 快照测试; 类别 docs; 类型 documentation) : 快照更新: GLM-4.5-Air (rotary_base=1000000) 的命令行新增 MODEL_ARGS_ROTARY_BASE=1000000 前缀, 记录修复后的真实行为。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/GLM-4.5-Air/p2p.txt (模块 快照测试; 类别 docs; 类型 documentation) : 快照更新: p2p 模式下同样的 rotary_base 前缀, 保证两种传输模式行为一致。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/Qwen3-235B-A22B-Instruct-2507/broadcast.txt (模块 快照测试; 类别 docs; 类型 documentation) : 快照更新: Qwen3-235B-A22B-Instruct-2507 (rotary_base=5000000) 的命令行新增 MODEL_ARGS_ROTARY_BASE=5000000 前缀。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/Qwen3-235B-A22B-Instruct-2507/p2p.txt (模块 快照测试; 类别 docs; 类型 documentation) : 快照更新: p2p 模式下 Qwen3-235B 的 rotary_base 前缀, 保持两种模式一致。
- tests/snapshots/launch_scripts/self_executing/examples/infra_features/p2p_weight_transfer/run.py/run/Qwen3-30B-A3B/broadcast.txt (模块 快照测试; 类别 docs; 类型 documentation) : 快照更新: Qwen3-30B-A3B (rotary_base=1000000) 的命令行新增 MODEL_ARGS_ROTARY_BASE=1000000 前缀。

关键符号: build_model_args_env, build_model_args_command, expand_model_args, test_model_args_env_is_empty_when_the_profile_pins_no_rotary_base, test_model_args_env_carries_the_rotary_base_a_profile_pins, test_a_pinned_rotary_base_reaches_the_expanded_model_args, test_every_profile_expands_to_a_usable_argv

关键源码片段

examples/infra_features/p2p_weight_transfer/run.py

核心修复文件: 新增 build_model_args_env / build_model_args_command, 把 rotary_base 注入 source 前的 shell 环境, 解决 profile 参数无法到达模型脚本的问题。

```
# examples/infra_features/p2p_weight_transfer/run.py (节选)
import shlex # 新增: 构建 shell 片段时需要取值做引用
```

```
def build_model_args_env(cfg: RunConfig) -> dict[str, str]:
    """把 profile 中的 rotary_base 统一转成模型脚本读取的 MODEL_ARGS_* 环境变量。

    返回空字典表示该 profile 不固定 rotary_base, 避免向两个注入点
```

传递错误的覆盖值。

```
"""
```

```
if cfg.rotary_base is None:
```

```
    return {}
```

```
return {"MODEL_ARGS_ROTARY_BASE": str(cfg.rotary_base)}
```

```
def build_model_args_command(cfg: RunConfig) -> str:
```

```
    """生成一个 shell 片段：先设置 MODEL_ARGS_*, 再 source 模型脚本。
```

```
    修复前 rotary_base 只写入 ray runtime env, 模型脚本在 shell 中展开
```

```
    ${MODEL_ARGS[@]} 时读不到它, 导致 --rotary-base 与 profile 声明不符。
```

```
    shlex.quote 用于防止值中的特殊字符破坏命令结构。
```

```
    """
```

```
    prefix = "".join(f"{name}={shlex.quote(value)} " for name, value in build_model_args_env(cfg).items())
```

```
    return f'{prefix}source "{MILES_ROOT}/scripts/models/{cfg.model_type}.sh"'
```

```
# cmd_run 中两个使用点（节选）：runtime env 与提交命令共用同一来源
```

```
model_args_env = build_model_args_env(cfg)
```

```
model_args_source = build_model_args_command(cfg)
```

```
# ...
```

```
env_vars.update(model_args_env) # ray 作业进程可见
```

```
runtime_env_json = json.dumps({"env_vars": env_vars})
```

```
# ...
```

```
run_cmd(
```

```
    f"{model_args_source} && "
```

```
    f"ray job submit --address='http://127.0.0.1:8265' "
```

```
    f"--runtime-env-json='{runtime_env_json}' "
```

```
    f"-- python3 \"{MILES_ROOT}/train.py" '
```

```
    f"${{MODEL_ARGS[@]}} " # 此刻已能看到 MODEL_ARGS_ROTARY_BASE
```

```
    f"{args_str}",
```

```
    check=False,
```

```
)
```

评论区精华

该 PR 的 review 讨论非常少：only 审核者 yueming-yuan 直接 Approve，未留下任何文字评论，说明变更被认为没有阻塞性问题。Issue 上仅有一条 gemini-code-assist[bot] 的公告（GitHub 上的 Gemini Code Assist 已停止 review 活动），不构成有效审查意见。整体决策由作者 fzyzcjy 主导，依托快照测试作为行为契约。

- reviewer 直接批准，无文字评论 (other): 变更被认为不存在阻塞性问题，直接批准并合并。

风险与影响

- 风险:

- 配置注入双点同步：rotary_base 现在同时注入 shell 前缀和 ray runtime env，两处共用 build_model_args_env 单一数据源，短时间不会漂移；但未来若新增其他 MODEL_ARGS_* 键，必须同步在该函数注册，否则会出现注入缺失。
- 快照依赖 bash 具体行为：快照测试和 expand_model_args 显式使用 /bin/bash，若执行环境 shell 语义不同会直接失败；这是显式保护而非隐患，但意味着修改模型脚本时必须同步重录快照。
- 示例级修复影响面有限：改动仅涉及 examples/infra_features/p2p_weight_transfer/run.py 及对应测试 / 快照，不影响 miles/ 核心库或 CI 门禁。风险集中在 P2P 权重传输示例复现正确性。
- 测试进程副作用：fast 测试通过 importlib 加载 run.py，若该文件今后引入重依赖或顶层副作用，可能拖慢 fast 套件或造成 pytest 导入异常。
- 影响：
 - 用户（示例使用者）：P2P 权重传输示例中 Qwen3-235B-A22B-Instruct-2507（rotary_base=5000000）、Qwen3-30B-A3B（1000000）、GLM-4.5-Air（1000000）等模型现在会真正以 profile 声明的 rotary base 训练，不再静默使用默认值。
 - 工程影响：为 #1837 推进的“启动脚本命令快照 + 模型参数展开”链条补齐关键一环；后续 #1908-#1911 都复用同一展开 / 快照路径，本 PR 是这条链路的行为基准之一。
 - 团队影响：新增一个测试目录（tests/fast/examples/...），不改变现有测试运行方式，fast 套件自动发现新用例。
 - 风险标记：配置注入双点同步风险，快照依赖 bash 具体行为，示例级修复不影响核心库

关联脉络

- PR #1911 Quote the model args miles inlines into the launch command: 同一功能链的后续：为内联 model args 加 shell 引号，与本 PR 的 shlex.quote 前缀构造直接相关，共同完善 MODEL_ARGS 传递链路。
- PR #1909 Expand the model args in python before building the command: 本 PR 之后对命令构建路径的进一步重构，把 model args 展开逻辑上移，与本 PR 的 build_model_args_command 形成演进关系。
- PR #1910 Replace the model config shell scripts with python: 同一启动脚本现代化系列：将模型配置 shell 脚本替换为 Python，与本 PR 操作的 source 模型脚本路径直接相关。
- PR #1908 Snapshot test the argv of all model scripts: 同批次快照基建扩展，将本 PR 建立的“真实 argv 快照 / 展开验证”模式推广到全部模型脚本。