

PR #1909 完整报告

radixark/miles

Expand the model args in python before building the command

合并时间: 2026-08-09 18:48

原文链接: <http://prhub.com.cn/radixark/miles/pull/1909>

执行摘要

- 一句话: Python 预展开模型参数, 命令构建不再依赖 shell 展开
- 推荐动作: 值得精读。该 PR 展示了三个可复用的技术点: 如何在 python 中安全地展开 shell 数组 (NUL 分隔 + 空白符断言); 如何通过环境变量冻结保证快照可复现; 以及如何用快照锁定重构前后的行为一致性。建议结合 #1908、#1910、#1911 一起阅读, 理解这条启动脚本重构链路的完整设计。

功能与动机

本 PR 是 tracking issue #1837 的一部分。其直接动机来自 #1908 的提交说明: "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, 在把模型配置脚本从 shell 重写为 python 之前, 需要先把每个脚本展开出的完整 argv 固化为快照, 作为重写是否忠实的唯一基准。同时, 预先在 python 侧展开参数也让最终提交到 ray 的命令完全确定、可读、可复现, 不受提交机与执行机环境差异影响。

实现拆解

实现分为 5 步, 均在 launch 脚本命令构建链路上:

1. 新增 `load_model_args` 函数 (`miles/utils/external_utils/command_utils.py`): 接受 `megatron_model_type`, 在 python 中通过 `subprocess.run(["bash", "-c", ...])` 执行 `source <model>.sh && printf "%s\\0" "${MODEL_ARGS[@]}"`, 用 NUL 分隔输出数组元素, 拆分后断言每个 token 不含空白符, 最后返回字符串列表。该函数在命令构建时 (而非运行时) 展开模型参数, 使后续构建的命令字符串不再依赖目标机器的 bash 环境。
2. 改造 `convert_checkpoint`: 原先命令以 `source <repo>/scripts/models/<type>.sh && ... ${MODEL_ARGS[@]}` 拼接; 现在改为 `' '.join(load_model_args(megatron_model_type))` 直接把展开后的参数内联进 `torchrun` 命令行, 并删掉 `source` 前缀。
3. 改造 `execute_train`: 原先在 `ray job submit` 命令前拼接 `source ... && ${MODEL_ARGS[@]}`; 现在同样预先展开为 `model_args` 字符串内联进提交命令。FSDP 路径 (`megatron_model_type=None`) 继续保持不展开任何模型参数。
4. 更新测试与测试框架: `tests/fast/utils/test_command_utils.py` 中原先断言 `source` 与 `${MODEL_ARGS[@]}` 出现的测试改为断言展开后的具体参数 (如 `--num-layers 36`), 并

确认命令中不再出现 `source`；FSDP 用例改为断言展开参数不存在。

`tests/fast/launch_scripts/py_harness.py` 中 `_CLEARED_ENV` 更名为 `CLEARED_ENV` 并新增 `MODEL_ARGS_NUM_LAYERS`、`MODEL_ARGS_ROTARY_BASE`、`MODEL_ARGS_FIRST_K_DENSE_REPLACE`、`ROTARY_SCALING_FACTOR` 等环境键，确保快照不受开发者本机环境覆盖影响。`tests/fast/launch_scripts/test_py_launch_scripts.py` 新增 `test_every_environment_knob_a_model_script_reads_is_frozen`，通过正则扫描所有 `scripts/models/*.sh` 中读取的 `${VAR:-}` 与 `environ.get("VAR")` 形式的环境变量，断言全部被 `CLEARED_ENV` 覆盖，防止未来新增环境变量破坏快照可复现性。

5. 重新生成全部 python 启动脚本快照：`tests/snapshots/launch_scripts/py/scripts/` 下所有 `full_train.txt`、`prepare.txt`、`train.txt` 等文件中的 `${MODEL_ARGS[@]}` 被替换为展开后的完整参数列表（如 DeepSeek-V4 的 40 余个 `--dsv4-*` 参数），涉及的模型包括 DeepSeek-V4、DeepSeek-V3.2、DeepSeek-V3、GLM-5、Qwen 系列、Joy-AI 等，共 20 余个快照文件。这些快照是后续 #1910 shell→python 重写的忠实性基线。

关键文件：

- `miles/utils/external_utils/command_utils.py`（模块 命令工具；类别 `source`；类型 `core-logic`；符号 `load_model_args`）：核心源码：新增 `load_model_args` 并用预展开参数替换 `convert_checkpoint` 与 `execute_train` 中的 `source ${MODEL_ARGS[@]}` 模式，是本次重构的主入口。
- `tests/fast/utils/test_command_utils.py`（模块 命令测试；类别 `test`；类型 `test-coverage`；符号 `test_expands_the_model_config_into_the_submitted_command`，`test_omits_the_model_args_for_fsdp`）：测试核心改动：将断言 `source/${MODEL_ARGS[@]}` 的用例改为断言展开后的具体参数，并新增 FSDP 遗漏检查，验证新行为。
- `tests/fast/launch_scripts/test_py_launch_scripts.py`（模块 启动脚本测试；类别 `test`；类型 `test-coverage`；符号 `test_every_environment_knob_a_model_script_reads_is_frozen`）：新增环境变量冻结测试，确保模型脚本读取的所有环境 knob 都被 `CLEARED_ENV` 覆盖，保障快照可复现性。
- `tests/fast/launch_scripts/py_harness.py`（模块 测试框架；类别 `test`；类型 `test-coverage`）：测试框架同步：`CLEARED_ENV` 更名为 `CLEARED_ENV` 并新增 `MODEL_ARGS*` 与 `ROTARY_SCALING_FACTOR` 等按键，避免开发机环境变量污染快照。
- `tests/snapshots/launch_scripts/py/scripts/run_deepseek_v4.py/full_train.txt`（模块 快照；类别 `docs`；类型 `documentation`）：代表性快照更新：将 `${MODEL_ARGS[@]}` 替换为展开后的 DeepSeek-V4 完整参数列表，是本次行为变化最直观的产物。

关键符号：`load_model_args`

关键源码片段

`miles/utils/external_utils/command_utils.py`

核心源码：新增 `load_model_args` 并用预展开参数替换 `convert_checkpoint` 与 `execute_train` 中的 `source ${MODEL_ARGS[@]}` 模式，是本次重构的主入口。

```
# miles/utils/external_utils/command_utils.py (PR#1909 新增)
```

```
def load_model_args(megatron_model_type: str) -> list[str]:
    """在 Python 侧展开 scripts/models/<megatron_model_type>.sh 声明的 MODEL_ARGS 数组。

    原实现把 `source <model>.sh && ${MODEL_ARGS[@]}` 原样拼进命令字符串，
    由目标机器的 bash 在运行时展开；本 PR 改为在构建命令前预先展开，
    使最终提交的 argv 完全确定、可快照、可复现。
    """
    script = f"{repo_base_dir}/scripts/models/{megatron_model_type}.sh"
    # 脚本不存在时立即失败，而不是让任务提交到远端后才报错
    assert os.path.exists(script), f"no model args script at {script}"

    # 用 NUL 分隔输出每个数组元素，避免空格分隔无法区分“含空格的一个 token”和“多个 token”
    expansion = f'source {shlex.quote(script)} && printf "%s\\0" "${{MODEL_ARGS[@]}}"'
    result = subprocess.run(["bash", "-c", expansion], capture_output=True, text=True, check=
    True)

    # 去掉末尾的空 token 就是完整列表
    tokens = result.stdout.split("\0")[:-1]

    # 每个 token 必须是一个不含空白符的单词，否则后续 ' '.join 后无法再还原成多个参数
    for token in tokens:
        assert token.split() == [token], f"model args token must be one whitespace-free word:
        {token!r}"
    return tokens
```

tests/fast/utils/test_command_utils.py

测试核心改动：将断言 `source/${MODEL_ARGS[@]}` 的用例改为断言展开后的具体参数，并新增 FSDP 遗漏检查，验证新行为。

tests/fast/utils/test_command_utils.py (PR#1909 更新)

```
def test_expands_the_model_config_into_the_submitted_command(self, commands):
    """模型配置不再以 `source` 形式出现在命令里，而是被展开为具体 argv。"""
    command_utils.execute_train(train_args="--x 1", num_gpus_per_node=8, megatron_model_
    type="qwen3-4B")

    submit = commands[-1]
    # 断言展开后的真实参数出现在命令中
    assert "--num-layers 36 " in submit
    # 不再依赖目标机器上的 bash 去 source 模型脚本
    assert "source" not in submit
    # 用户自定义参数仍保持在末尾
    assert submit.endswith("--x 1")
```

评论区精华

该 PR 未产生实质评讨论。收到的唯一 review 是 yueming-yuan 的 APPROVED，无评论内容；关联 Issue 中仅有一条 `gemini-code-assist[bot]` 的自动提示，声明其代码审查服务已停止，不构成有效评审。因此本 PR 的决策主要依赖作者 (fzyzcjy) 在 squash commit 中的设计

说明和测试覆盖。

- 暂无高价值评论线程

风险与影响

- 风险:

1. 环境差异风险: `load_model_args` 在 `python` 进程内通过 `bash -c` 执行模型脚本并展开参数, 如果模型脚本依赖运行时动态环境 (如 `SLURM_JOB_NUM_NODES`、`ROTARY_SCALING_FACTOR` 等), 展开结果会随调用进程的环境而变化, 可能与以往在目标执行机上展开的行为不一致。新增的环境冻结测试只覆盖了当前脚本中出现的按键, 未来新增按键可能暂时漏网。
2. 空白符断言失效风险: `load_model_args` 强制每个 token 不含空白符, 如果未来某个模型参数值需要包含空格 (如带空格的文件路径), 函数会直接断言失败, 导致所有使用该模型类型的 `launch` 脚本无法构建命令。
3. 子进程开销: 每次构建命令都额外 `spawn` 一个 `bash` 子进程, 虽然频率不高 (仅启动时), 但在多节点大量并行提交时会有轻微 CPU 开销。
4. 快照规模膨胀: 预展开使每个快照文件增加数十行参数, 52 个变更文件中有大量为快照更新, 任何模型脚本参数变更都会导致快照 `diff` 变大, 评审负担加重。
5. FSDP 路径独立性: FSDP 分支不展开模型参数, 逻辑维持原样, 但测试已改为断言展开参数不存在, 避免误回归。- 影响: 影响所有通过 `miles.utils.external_utils.command_utils` 构建 `ray` 提交命令的 `python` 启动脚本 (`scripts/` 与 `examples/` 下的 `run_*.py`), 以及依赖这些命令的 CI 快照测试。用户可见的变化是: 提交到 `ray` 的完整命令现在直接展示全部模型参数, 日志排查与复现更容易; 快照测试从断言 `shell` 占位符变为断言具体展开值, 任何模型脚本改动都会立刻在 CI 中反映。该 PR 为 #1910 (`shell`→`python` 重写) 提供了行为基线, 是后续重构的关键前置步骤。- 风险标记: 核心启动路径变更, 引入子进程调用, 快照同步更新, 环境变量敏感性

关联脉络

- PR #1910 Replace the model config shell scripts with python: 该 PR 是 #1909 的下游步骤: 在 `shell` 版本被删除前, 本 PR 固化了每个模型的展开 `argv`, 为 `shell`→`python` 重写提供行为基线。
- PR #1911 Quote the model args miles inlines into the launch command: 本 PR 将展开后的参数以 `' '.join` 直接内联, 后续 PR#1911 进一步为内联参数加 `shell` 引号, 修复 `MoE` 数组参数被 `glob` 展开的问题, 两者紧密衔接。
- PR #2279 Run the launch script snapshot tests by hand instead of in CI: 本 PR 引入的大量快照文件, 后来被 #2279 移出 CI 改为手动执行, 涉及同一批 `tests/snapshots` 资产的维护方式演进。