

PR #27538 完整报告

sgl-project/sglang

[AMD] ci: reinstall MoRI if Different from Dockerfile-pinned commit during install_dependency

合并时间: 2026-06-08 16:29

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27538>

执行摘要

- 一句话: CI 中动态重装 MoRI 以匹配 Dockerfile 版本
- 推荐动作: 该 PR 是基础设施层面的维护, 值得关注其解决特定 CI 问题的方法: 在不可变 CI 镜像中动态替换预装组件以匹配当前代码。对于维护 CI 管线的工程师, 可借鉴其模式。一般工程师无需深入阅读。

功能与动机

AMD CI 镜像预装了 MoRI, 但不会因 PR 变更镜像。当 PR#26922 更新 `MORI_COMMIT` 以引入新 API (如 `IOEngine.wait_all`) 时, CI 容器仍使用旧 MoRI, 导致反聚合测试失败。需在依赖安装阶段动态重装 MoRI 以匹配 Dockerfile。

实现拆解

在 `scripts/ci/amd/amd_ci_install_dependency.sh` 文件末尾, `accelerate` 安装之后, 插入一段条件逻辑:

1. 通过 `docker exec ci_sglang test -d /sgl-workspace/mori` 检查容器内是否存在 MoRI 目录 (仅 `ENABLE_MORI=1` 的镜像包含)。
2. 如果存在, 则从 `docker/rocm.Dockerfile` 中提取 `MORI_REPO` 和 `MORI_COMMIT` 变量。
3. 根据 `GPU_ARCH` 环境变量设置 MoRI 的 GPU 架构 (`gfx950` 或 `gfx942`)。
4. 在容器内依次执行: 删除旧 MoRI 目录、克隆指定仓库、切换到指定 commit、初始化子模块、运行 `python3 setup.py develop` 安装、配置 `torch` 库路径并运行 `ldconfig`。
5. 整个过程记录日志, 并在完成后输出 `[MORI] Done.`。该步骤无额外配置或测试配套。

关键文件:

- `scripts/ci/amd/amd_ci_install_dependency.sh` (模块 部署脚本; 类别 `infra`; 类型 `infrastructure`): 该文件是唯一的变更文件, 包含在 CI 容器内动态重装 MoRI 的完整脚本逻辑, 是 PR 的核心。

关键符号: 未识别

关键源码片段

`scripts/ci/amd/amd_ci_install_dependency.sh`

该文件是唯一的变更文件，包含在 CI 容器内动态重装 MoRI 的完整脚本逻辑，是 PR 的核心。

```
# 脚本片段: scripts/ci/amd/amd_ci_install_dependency.sh (新增部分)

# 检查容器内是否存在 MoRI 目录 (仅 ENABLE_MORI=1 的镜像包含)
if docker exec ci_sglang test -d /sgl-workspace/mori; then
    # 从 docker/rocm.Dockerfile 中提取 MORI_REPO 和 MORI_COMMIT
    MORI_REPO=$(grep -E '^[:space:]*ARG[:space:]+MORI_REPO=' docker/rocm.Dockerfile |
    head -n1 | sed 's/.*MORI_REPO="\([^"]*\)".*/\1/')
    MORI_COMMIT=$(grep -E '^[:space:]*ARG[:space:]+MORI_COMMIT=' docker/rocm.
    Dockerfile | head -n1 | sed 's/.*MORI_COMMIT="\([^"]*\)".*/\1/')

    # 根据 GPU_ARCH 设置 MoRI 的 GPU 架构
    if [[ "${GPU_ARCH}" == "mi35x" ]]; then
        MORI_GPU_ARCHS="gfx950"
    else
        MORI_GPU_ARCHS="gfx942"
    fi

    echo "[MORI] Reinstalling MORI ${MORI_COMMIT} (MORI_GPU_ARCHS=${MORI_GPU_ARCHS})"

    # 在容器内重新安装 MoRI: 删除旧目录、克隆、切换 commit、初始化子模块、安装
    docker exec ci_sglang bash -c "
        set -euo pipefail
        export MORI_GPU_ARCHS='${MORI_GPU_ARCHS}'
        rm -rf /sgl-workspace/mori
        git clone '${MORI_REPO}' /sgl-workspace/mori
        cd /sgl-workspace/mori
        git checkout '${MORI_COMMIT}'
        git submodule update --init --recursive
        python3 setup.py develop
        python3 -c 'import os, torch; print(os.path.join(os.path.dirname(torch.__file__), 'lib'))' > /
        etc/ld.so.conf.d/torch.conf
        ldconfig
    "
    echo "[MORI] Done."
fi
```

评论区精华

机器人评论指出，直接将宿主变量嵌入双引号 `docker exec bash -c "..."` 可能存在 shell 转义风险，建议使用 `docker exec -e` 传递环境变量并用单引号包裹命令。该建议未在后续讨论中被采纳或拒绝，PR 已被合并。

- host 变量传递的安全性 (design): 未采纳建议；PR 已合并，代码保持原有方式。

风险与影响

- 风险：风险较低。该变更仅影响 AMD CI 中启用了 MoRI 的容器（通过 /sgl-workspace/mori 目录守卫），不影响其他套件。潜在问题包括：
 - 如果 Dockerfile 中 MORI_COMMIT 或 MORI_REPO 格式变化，提取逻辑可能失败。
 - 重装 MoRI 会增加依赖安装步骤的时间。
 - 未处理网络问题导致 clone 失败的情况。
 - 影响：影响范围限于 AMD CI 中启用了 MoRI 的测试套件（主要是 mi35x 反聚合测试）。该步骤确保 CI 使用与 Dockerfile 完全一致的 MoRI 版本，避免因镜像陈旧导致的测试失败。对其他 GPU 架构或无 MoRI 的套件无影响。
- 风险标记：低风险基础设施变更，缺少网络异常处理

关联脉络

- PR #26922 bump MORI_COMMIT in docker/rocm.Dockerfile: 本 PR 正是为了配合 #26922 的变更而提出的，确保 CI 容器使用更新后的 MoRI commit。