

PR #27860 完整报告

sgl-project/sglang

ci(xpu): pull intel/sglang-dev:latest and clean workspace properly

合并时间: 2026-06-11 13:14

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27860>

执行摘要

- 一句话: 重构 XPU CI 容器镜像拉取策略并修复权限问题
- 推荐动作: 建议合并此 PR。虽然没有功能性的修改, 但显著提升了 XPU CI 的健壮性和可维护性。值得关注的设计决策是: 直接依赖 :latest 标签权衡了便利性与一致性, 团队应确保发布工作流打 :latest 标签的时机合适且明确。

功能与动机

PR body 指出: 发布工作流已经将 :latest 标签与 nightly 日期标签一起推送, 因此无需再回退扫描历史标签; pip install 以 root 身份在容器内运行, 导致挂载的工作空间内文件归 root 所有, 主机清理时 rm 因权限不足失败 (被 ll true 静默忽略)。此外, 为了更完整地清理构建产物, 需要扩大 rm 范围。

实现拆解

实现了三项主要变更:

1. 简化容器镜像拉取 (scripts/ci/xpu/xpu_ci_start_container.sh): 用直接 pull intel/sglang-dev:latest 替换原先遍历近 7 天 nightly 标签的复杂逻辑; 新增 --image-tag 参数替代 --lookback-days, 允许调用方仍然指定特定标签。
2. 修复工作空间清理权限 (pr-test-xpu.yml 两阶段 cleanup): 在 stage-a 和 stage-b 的清理步骤中, 先用 docker run busybox chown 将工作空间内的文件所有者变更为 runner 用户, 再执行 rm, 避免因 root 所有权导致清理失败。
3. 扩展清理范围并启用 --continue-on-error: 新增 python/dist、*.egg-info、test/results、.pytest_cache、__pycache__ 等目录的清理; 在 stage-b 的 run_suite.py 调用中添加 --continue-on-error, 保证即使某个测试文件失败也能完成全部测试并输出摘要。没有涉及源代码或测试的修改, 全部为 CI 基础设施脚本和工作流的优化。

关键文件:

- scripts/ci/xpu/xpu_ci_start_container.sh (模块 CI 脚本; 类别 infra; 类型 infrastructure): 核心变更点: 重构容器镜像拉取策略, 移除复杂的按日期回退逻辑, 直接使用 :latest 标签, 并替换参数接口。
- .github/workflows/pr-test-xpu.yml (模块 CI 工作流; 类别 infra; 类型 infrastructure): CI 工作流文件, 增加工作空间清理的权限修正、扩展清理范围, 并在 stage-b 测试中添加 --continue-on-error 选项。

关键符号：未识别

关键源码片段

scripts/ci/xpu/xpu_ci_start_container.sh

核心变更点：重构容器镜像拉取策略，移除复杂的按日期回退逻辑，直接使用 :latest 标签，并替换参数接口。

```
#!/bin/bash
set -euo pipefail

# Start the Intel XPU CI container (ci_sglang_xpu) using the intel/sglang-dev:latest
# image published by .github/workflows/release-docker-intel-xpu-nightly.yml.
# Pulls the :latest tag and starts a long-running container that subsequent
# steps `docker exec` into.

CONTAINER_NAME="ci_sglang_xpu"
IMAGE_REPO="intel/sglang-dev"
IMAGE_TAG="latest" # 直接使用 latest 标签，不再回退扫描日期标签
CUSTOM_IMAGE=""

while [[ $# -gt 0 ]]; do
  case $1 in
    --custom-image) CUSTOM_IMAGE="$2"; shift 2;;
    --container-name) CONTAINER_NAME="$2"; shift 2;;
    --image-tag) IMAGE_TAG="$2"; shift 2;; # 替代原有 --lookback-days 参数
    -h|--help)
      echo "Usage: $0 [OPTIONS]"
      echo "Options:"
      echo "  --custom-image IMAGE   Use a specific Docker image directly"
      echo "  --container-name NAME  Override container name (default: ${CONTAINER_NAME})"
      echo "  --image-tag TAG        Tag of ${IMAGE_REPO} to pull (default: ${IMAGE_TAG})"
      exit 0 ;;
    *) echo "Unknown option $1"; exit 1;;
  esac
  shift
done

# ... 后续拉取并启动容器的逻辑基本保持不变 ...
```

评论区精华

review 简洁，审核人 mingfeima 直接批准通过，无其他评论讨论。

- 暂无高价值评论线程

风险与影响

- 风险：本次变更为 CI 基础设施优化，风险较低。主要潜在风险：

- 环境一致性：直接使用 :latest 标签可能导致不同时间运行 CI 的镜像版本不一致，如果 :latest 更新了不兼容的依赖，可能引入难以追踪的失败。但发布工作流同时保留了日期标签，必要时可通过 --image-tag 锁定特定版本。
- 清理覆盖完整性：虽然扩展了清理范围，如果未来的构建产出新增目录或文件类型未在 rm 列表中，仍可能遗留文件影响下次运行。
- --continue-on-error 影响：虽然有利于全面报告，但可能让运维人员忽略关键失败，需要结合测试结果报告仔细检查。
- 影响：
- 用户影响：无。变更仅限 CI 基础设施。
- 系统影响：Intel XPU CI 构建流程更稳定、清理更彻底，减少了因上一轮构建残留导致的随机失败。
- 团队影响：镜像拉取速度提升（跳过回退扫描），测试结果更完整（continue-on-error），开发人员调试 CI 失败体验改善。
- 风险标记：CI 基础设施变更，容器镜像标签简化，工作空间清理权限修复

关联脉络

- 暂无明显关联 PR