

PR #23723 完整报告

sgl-project/sglang

[CI] sgl-kernel: prune dangling images before each wheel build

合并时间: 2026-04-26 01:44

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/23723>

执行摘要

- 一句话: 在 CI 构建前清理 Docker dangling 镜像和卷
- 推荐动作: 建议合并。这是一项低风险的 CI 基础运维改进, 直接解决已观测到的磁盘占满问题。

功能与动机

CI 自托管 runner 磁盘因 sgl-kernel 构建累积的 dangling 镜像 (68 个、约 480 GB) 和孤立的 buildx 卷 (5 个、约 38 GB) 被占满, 导致构建任务因磁盘空间不足而失败。PR body 引用了 PR #23720 的具体构建失败实例。

实现拆解

1. 在 `sgl-kernel-build-wheels` job 中添加清理步骤: 文件 `.github/workflows/pr-test.yml`, 第 482-504 行。新增 Free Docker disk space step, 首先执行 `docker image prune -f --filter "until=12h"` 删除 12 小时以上的 dangling 镜像, 避免与同一矩阵中其他单元格 (如不同 CUDA 版本) 刚产生的 dangling 镜像冲突。然后遍历 `docker volume ls` 中名为 `buildx_buildkit_*` 的卷, 排除正在使用的 `buildx_buildkit_sgl-kernel-builder`, 删除其余孤立卷。最后执行 `df -h /` 打印磁盘空间。
2. 在 `sgl-kernel-build-wheels-arm` job 中同样添加: 第 568-578 行, 完全相同的清理逻辑, 确保 arm 构建工作流也受保护。
3. 明确不清理的内容: `~/.cache/sgl-kernel/buildx` (构建缓存)、`~/.cache/sgl-kernel/ccache` (编译器缓存) 和带有 tag 的 `sgl-kernel-deps:*` 镜像被保留, 确保增量构建速度不变。

关键文件:

- `.github/workflows/pr-test.yml` (模块 CI 配置; 类别 infra; 类型 infrastructure) : CI 工作流文件, 新增磁盘清理步骤, 是变更的唯一生效位置。

关键符号: 未识别

关键源码片段

[.github/workflows/pr-test.yml](#)

CI 工作流文件, 新增磁盘清理步骤, 是变更的唯一生效位置。

```

# .github/workflows/pr-test.yml ( 片段 )
# 在 sgl-kernel-build-wheels job 中, 构建 wheel 之前
- name: Free Docker disk space
  run: |
    set -x
    # build.sh 每次运行都会重新标记 deps 镜像, 旧镜像变成 dangling 状态 (约 16-23 GB 每个) 。
    # 此处清理 12 小时以上的 dangling 镜像, 避免与并行矩阵单元格冲突。
    docker image prune -f --filter "until=12h"
    # 删除孤立的 buildx 构建器卷, 排除当前正在使用的 sgl-kernel-builder。
    for v in $(docker volume ls -q | grep '^buildx_buildkit_' | grep -v '^buildx_buildkit_sgl-kernel-
builder' || true); do
      echo "Removing orphan buildx volume: $v"
      docker volume rm "$v" || true
    done
    # 打印磁盘空间用于日志调试。
    df -h /

# .github/workflows/pr-test.yml ( 片段 )
# 在 sgl-kernel-build-wheels-arm job 中, 构建 wheel 之前
- name: Free Docker disk space
  run: |
    set -x
    # 参见上方 sgl-kernel-build-wheels 中的解释。
    docker image prune -f --filter "until=12h"
    for v in $(docker volume ls -q | grep '^buildx_buildkit_' | grep -v '^buildx_buildkit_sgl-kernel-
builder' || true); do
      echo "Removing orphan buildx volume: $v"
      docker volume rm "$v" || true
    done
    df -h /

```

评论区精华

讨论主要围绕 `until` 过滤器的时长选择。初始提交使用 1 小时, 在 review 中建议改为 12 小时以更保守地防止与同一矩阵其他单元格的竞争条件。最终第二笔提交将过滤器改为 `until=12h`, 并添加了 Co-Authored-By 署名。

- `until` 过滤器时长选择 (design): 第二笔提交改为 `until=12h`, 并添加 Co-Authored-By 署名。

风险与影响

- 风险: 无显著技术风险。清理操作仅针对 `dangling` 镜像和孤立构建卷, 不影响缓存和使用的镜像。但若 CI 运行间隔极短 ($<12h$), 且同时有大量不同 CUDA 版本的构建在并行执行, 可能清理不及时导致磁盘空间仍逐渐积累——不过 PR 设计已通过 12 小时窗口缓解此风险。
- 影响: 直接影响自托管 runner 的磁盘空间管理, 消除因空间不足导致的构建失败。对用户无感知, 对开发者减少 CI 运维干扰。影响范围限于 CI workflow 中的 `sgl-kernel-build-wheels` 和 `sgl-kernel-build-wheels-arm` 两个 job。

- 风险标记：无显著风险

关联脉络

- PR #23720（推断为触发本 PR 的构建失败）：PR body 引用该 PR 的构建失败作为磁盘空间问题的实例。