

PR #25206 完整报告

sgl-project/sglang

ci(sgl-kernel-build): actually reclaim disk in the wheel-build cleanup step

合并时间: 2026-05-14 09:45

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25206>

执行摘要

- 一句话: 修复 sgl-kernel 构建 CI 磁盘回收逻辑
- 推荐动作: 该 PR 为 CI 基础设施修复, 代码改动量小但修复了两个隐蔽的 bug, 值得相关维护者了解。建议关注后续 CI 运行中磁盘清理步骤是否打印清理信息 (如 "Removing orphan buildx builder" 行), 以确认修复生效。

功能与动机

自托管 runner 在执行 sgl-kernel 构建时因磁盘空间不足 (no space left on device, 写 libtorch_cuda.so 时) 导致构建失败。原磁盘清理步骤由于两个 bug 实际未生效: docker image prune -f --filter "until=12h" 因过滤器误排除了真正需要清理的上一次运行 dangling 镜像; docker volume rm 因 volume 被 buildkit helper 容器持有而失败并被 ll true 静默吞掉。另外本地 buildx 缓存 (~/.cache/sgl-kernel/buildx) 无上限增长, 在 labubu 上已达 726GB。

实现拆解

修改 [.github/workflows/_pr-test-sgl-kernel-build.yml](#) 中的 "Free Docker disk space" 步骤:

1. Docker image prune: 移除 --filter "until=12h", 仅用 -f (不加 -a), 只清理 dangling 镜像 (上次构建留下的 <none>:<none>), 保留带 tag 的 sgl-kernel-deps:* 基础镜像。
2. 孤儿 builder 清理: 用 docker buildx ls --format '{{.Name}}' | grep -E '^builder-' 列出所有 orphan 的 builder, 逐一执行 docker buildx rm —— 这会同时删除 builder 对应的 helper 容器和其持有的 volume, 避免之前 docker volume rm 总是失败的问题。保留了 sgl-kernel-builder 和 default builder。
3. buildx 缓存裁剪: 新增对 ~/.cache/sgl-kernel/buildx/blobs 按 mtime 超过 14 天进行 find -type f -mtime +14 -delete, 因为 cache-hit 导入会刷新 blob 的 mtime, 所以最近构建依赖的层不会被误删。
4. 并发安全说明: 两个 matrix cell 共享缓存路径, 但基于 content-addressed 存储, find -delete 按 inode 删除, POSIX unlink-while-open 保证并发读取安全; 最坏情况是已删除 blob 触发重新拉取, 不会导致数据损坏。

关键文件:

- [.github/workflows/_pr-test-sgl-kernel-build.yml](#) (模块 CI 配置; 类别 infra; 类型 infrastructure): 核心变更文件, 修复了磁盘清理步骤中的两个 bug (prune 过滤器和

volume rm 失败) , 并新增了 buildx 缓存裁剪逻辑。

关键符号: 未识别

关键源码片段

[.github/workflows/_pr-test-sgl-kernel-build.yml](#)

核心变更文件, 修复了磁盘清理步骤中的两个 bug (prune 过滤器和 volume rm 失败) , 并新增了 buildx 缓存裁剪逻辑。

```
# .github/workflows/_pr-test-sgl-kernel-build.yml (cleanup 步骤关键片段)
- name: Free Docker disk space
  run: |
    set -x
    # `f` only (no `a`): drops dangling <none>:<none> layers from
    # prior runs but keeps the tagged sgl-kernel-deps:* base images.
    docker image prune -f

    # Reap orphan buildx builders from interrupted prior runs. Must
    # `buildx rm` (not `volume rm`) — the volume is held open by the
    # buildkit helper container; removing the builder frees both.
    # `sgl-kernel-builder` and `default` are skipped on purpose.
    for b in $(docker buildx ls --format '{{.Name}}' | grep -E '^builder-' || true); do
      echo "Removing orphan buildx builder: $b"
      docker buildx rm "$b" || true
    done

    # Trim the persistent local buildx cache (--cache-to mode=max grows
    # unbounded). Cache-hit imports rewrite the blob, so mtime tracks
    # "last carried forward by a build" — 14d is well past the freshest
    # CI cycle.
    if [ -d "$HOME/.cache/sgl-kernel/buildx/blobs" ]; then
      find "$HOME/.cache/sgl-kernel/buildx/blobs" -type f -mtime +14 -delete
    fi

    df -h /
```

评论区精华

无 review 讨论 (无 review 评论) 。

- 暂无高价值评论线程

风险与影响

- 风险:
 - 回归风险: 如果某次构建恰好在 14 天以上未运行, 其缓存层可能被误删, 导致重新拉取缓存耗时增加。但由于 cache-hit 会刷新 mtime, 这种情况在正常 CI 周转下不会发生。

- 并发风险：两个 matrix cell 共享 ~/.cache/sgl-kernel/buildx, find -delete 在并发读写下可能删除正在被写入的 blob，但基于 content-addressed 和 POSIX unlink-while-open 语义，最多触发重拉取，不导致数据损坏。
- 误删风险：docker buildx rm 仅删除 ^builder- 前缀的 builder, sgl-kernel-builder 和 default 被排除，不会影响正常使用。
- 影响：直接影响 sgl-kernel 构建 CI 流程，避免 runner 节点因磁盘满载导致的构建失败。
间接影响：减少了磁盘空间使用（实验释放约 400GB），可能提高 runner 可用性和构建成功率。不影响用户产品代码。
- 风险标记：缓存裁剪可能误删，并发安全依赖 POSIX 语义

关联脉络

- PR #25135 ci: merge sgl-kernel-build-wheels x86+arm into reusable workflow: 本 PR 修改的正是 PR #25135 合并后引入的 _pr-test-sgl-kernel-build.yml workflow，磁盘清理 bug 在其基础上暴露。
- PR #24978（未提供具体标题，但在 body 中引用）：PR body 中提及该磁盘问题在 run 25786111587 / job 75753012008 (PR #24978) 中首次被发现。