

PR #51923 完整报告

vllm-project/vllm

[CI/Release][XPU] fix workdir path for triton shim job

合并时间: 2026-08-12 14:41

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51923>

执行摘要

- 一句话: 修复 XPU Triton 发布脚本临时目录路径, 适配 Docker 挂载
- 推荐动作: 值得快速阅读, 作为 51759 的后续修复, 理解 Docker 挂载与相对路径转换的小技巧; 不值得深度精读。关注点在于 shell 脚本中临时目录与容器挂载的兼容性处理。

功能与动机

PR body 直接给出了失败的 Buildkite 构建链接, 目标就是修复 release-v2 流水线中的 triton shim 发布任务。代码注释说明了根因: Docker Python fallback 会把仓库挂载到 `/app`, 在系统临时目录 (如 `/tmp`) 生成的文件在容器内不可见, 因此必须把临时目录放在仓库内, 并转为相对路径供容器访问。

实现拆解

1. 变更入口: `.buildkite/scripts/xpu/publish-triton-shim.sh`, 这是 XPU Triton shim 索引发布脚本, 位于 `.buildkite/scripts/xpu/`。
2. 替换临时目录创建: 由 `work_dir=$(mktemp -d)` 改为 `work_dir=$(mktemp -d "$PWD/.xpu-triton-index.XXXXXX")`, 确保临时目录位于仓库根目录下, Docker 挂载后可见。
3. 新增路径转换: `work_dir=${work_dir#"$PWD"/}`, 去掉仓库根前缀, 将绝对路径转为相对路径; 由于脚本开头已 `cd` 到仓库根, 相对路径在宿主机和容器 `/app` 下语义一致。
4. 保留清理逻辑: `trap 'rm -rf "$work_dir"' EXIT` 不变, 退出时仍会清理临时目录; `wheel_path="$work_dir/$WHEEL_FILENAME"` 无需改动, 但路径语义变为仓库相对路径。
5. 验证配套: 无新增测试, 依赖 CI (Buildkite #83516) 和后续 release 流水线验证; 改动不涉及其他配置或构建脚本。

关键文件:

- `.buildkite/scripts/xpu/publish-triton-shim.sh` (模块 发布脚本; 类别 `infra`; 类型 `bugfix`): 该脚本是 XPU Triton shim 索引发布流程的入口, PR 在此修复临时目录位置, 使其在 Docker 挂载仓库到 `/app` 后仍可访问, 是本次变更的核心文件。

关键符号: 未识别

关键源码片段

`.buildkite/scripts/xpu/publish-triton-shim.sh`

该脚本是 XPU Triton shim 索引发布流程的入口，PR 在此修复临时目录位置，使其在 Docker 挂载仓库到 /app 后仍可访问，是本次变更的核心文件。

```
# 进入仓库根目录，确保后续相对路径稳定
cd "$(dirname "${BASH_SOURCE[0]}")/../../.."

# 临时目录必须创建在仓库内：Docker Python fallback 会把仓库挂载到 /app，
# 系统临时目录（如 /tmp）在容器内不可见，会导致后续步骤读不到 wheel 文件。
work_dir=$(mktemp -d "$PWD/.xpu-triton-index.XXXXXX")

# 去掉仓库根前缀，把绝对路径转成相对路径，Docker 内 /app 下同样成立
work_dir=${work_dir#"$PWD"/}

# 退出时无论成功失败都清理临时目录，避免污染 git 仓库
trap 'rm -rf "$work_dir"' EXIT

wheel_path="$work_dir/$WHEEL_FILENAME"
```

评论区精华

无实质性技术讨论。claude[bot] 提示该 PR 来自 fork，自动审查被禁用；bigPYJ1151 直接批准且无评论，review 评论为空。核心信息都体现在代码注释中：临时目录必须放在仓库内以兼容 Docker Python fallback 的 /app 挂载。

- 暂无高价值评论线程

风险与影响

- 风险：风险具体在两点：一是若 CI 环境在脚本执行前重新定义 PWD，`work_dir=${work_dir#"$PWD"/}` 可能剥离失败，但脚本已显式 `cd` 到仓库根，此风险很低；二是若构建进程被强杀，EXIT trap 可能不执行，仓库根会残留 `xpu-triton-index.xxxxxx` 垃圾目录，但 CI 临时工作区或 `git clean` 通常能掩盖此问题。并发布由 `mktemp -d` 随机后缀保证互不冲突。
- 影响：影响范围仅限于 XPU 发布流水线（`.buildkite/scripts/xpu/publish-triton-shim.sh`），对模型推理、服务端代码零影响，对用户无感知。对发布工程师而言，修复了 `release 5061` 类失败，属于低风险、低影响的维护性修复。
- 风险标记：发布流水线脚本，无自动化测试覆盖，依赖 shell 路径剥离

关联脉络

- PR #51759 [CI/Release][1/N][XPU] Publish XPU Triton shim index: 该 PR 首次引入 `publish-triton-shim.sh` 和 `release-pipeline.yaml`，本 PR 修复其中工作目录路径问题，属于同一发布流程的后续修复。
- PR #51905 [XPU][CI] Change to use global `VLLM_DISABLE_COMPILE_CACHE=1` in Intel GPU CI: 同属 Intel GPU/XPU CI 维护系列，共享同一 CI 脚本目录和团队，存在功能线关联。