

PR #51666 完整报告

vllm-project/vllm

[CI][AMD] Persist the openai-harmony tiktoken vocab cache across jobs

合并时间: 2026-08-11 14:11

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51666>

执行摘要

- 一句话: AMD CI 持久化 harmony vocab 缓存并放宽文件模式校验
- 推荐动作: 值得快速阅读。重点看两个设计: 一是用 TIKTOKEN_RS_CACHE_DIR 把第三方运行时缓存重定向到持久卷、并让容器内外路径保持一致的写法; 二是 `core.fileMode=false` + `git diff` 退出码分段, 区分内容差异与 `git` 执行失败, 避免共享工作区文件模式漂移误杀 CI。对维护 AMD/ROCm CI 的同学尤其有价值。

功能与动机

PR body 记录了故障实例: amd-ci #11876 中 `mi355_1: Entrypoints Integration (API Server OpenAI - Part 2)` 上 8 个 `TestGPTOSSChat` 用例 4 个失败, 报 `openai_harmony.HarmonyError: error downloading or loading vocab file`。
`openai-harmony` 不内置 tiktoken vocab, 首次使用时下载并缓存在 `$TMPDIR/tiktoken-rs-cache`, 而 AMD 原生 CI 的 `TMPDIR` 是 job 级路径, 导致每个 job 都在请求时从 `openaipublic.blob.core.windows.net` 重新下载且无重试, 任何网络抖动都会变成测试中途的 500。作者强调这不是某个 PR 引入的回归, 惰性下载自 #22340 就存在, 只是切换原生执行后才让缓存成为可能。`ci-bake-rocm.sh` 的改动动机来自共享 AMD workspace 的文件可执行位漂移导致 CI Docker context 校验误报。

实现拆解

1. 重定向缓存根目录 (`.buildkite/scripts/hardware_ci/run-amd-test.sh` 的 `initialize_native_environment()`): 新增 `TIKTOKEN_RS_CACHE_DIR="${HF_HOME}/tiktoken-rs-cache"`, 并纳入 `export` 与 `mkdir -p` 预创建清单。`HF_HOME` 由脚本保证指向持久卷, 这样缓存生命周期从单 job 变为跨 job。
2. 容器内外路径一致: 在 `CONTAINER_PREFLIGHT` 的 `mkdir -p` 中加入 `$TIKTOKEN_RS_CACHE_DIR`, 并在 `docker run` 参数中注入 `-e "TIKTOKEN_RS_CACHE_DIR=${HF_MOUNT}/tiktoken-rs-cache"`, 确保容器内 `openai-harmony` 进程与宿主机共用同一持久缓存。由于缓存文件名由下载 URL 派生 (正文给出哈希 `fb374d...`), `vocab` 版本更新会自动换 key, 无需手动清理。
3. 修复 ROCm 构建上下文校验 (`.buildkite/scripts/ci-bake-rocm.sh` 的 `validate_ci_build_context_source()`): 将 `git diff` 改为 `git -c core.fileMode=false diff`, 只忽略共享工作区可执行位漂移, 仍在有真实内容变化时拒绝构建上下文; 同时用 `worktree_diff_status / staged_diff_status` 分段处理退出码, `>1` 视为 `git` 执行失败并输出

错误，1 时打印前 50 条差异供排障。

4. 验证与测试配套：未新增自动化测试文件，按 Test Plan 做了 bash -n 语法检查、确认缓存位置跟随 TMPDIR、以及打印缓存文件哈希验证版本换 key 三项目手动验证。

关键文件：

- .buildkite/scripts/hardware_ci/run-amd-test.sh (模块 硬件 CI; 类别 infra; 类型 infrastructure; 符号 initialize_native_environment, CONTAINER_PREFLIGHT) : 核心改动文件：通过 TIKTOKEN_RS_CACHE_DIR 将 openai-harmony 的 tiktoken vocab 缓存从 job 级 TMPDIR 重定向到持久卷 HF_HOME，并在容器 preflight 与 docker run 环境变量中透传，直接修复每 job 重复下载导致的 500 失败。
- .buildkite/scripts/ci-bake-rocm.sh (模块 构建脚本; 类别 other; 类型 core-logic; 符号 validate_ci_build_context_source) : 第三个提交加入的修复：
validate_ci_build_context_source() 用 core.fileMode=false 忽略共享工作区可执行位漂移，并分段处理 git diff 退出码，避免 ROCm CI Docker context 校验误报，与 PR 的 CI 稳定性主题一致。

关键符号：initialize_native_environment, validate_ci_build_context_source

关键源码片段

.buildkite/scripts/hardware_ci/run-amd-test.sh

核心改动文件：通过 TIKTOKEN_RS_CACHE_DIR 将 openai-harmony 的 tiktoken vocab 缓存从 job 级 TMPDIR 重定向到持久卷 HF_HOME，并在容器 preflight 与 docker run 环境变量中透传，直接修复每 job 重复下载导致的 500 失败。

```
# 把 openai-harmony 的 tiktoken vocab 缓存指到持久卷 HF_HOME
# 默认行为是缓存在 $TMPDIR 下，而 AMD CI 的 $TMPDIR 是 job 级的
# 因此每个 job 都要在请求时重新下载 vocab 并可能中途失败
TIKTOKEN_RS_CACHE_DIR="${HF_HOME}/tiktoken-rs-cache"
: "${HF_HUB_DOWNLOAD_TIMEOUT:=300}"
: "${HF_HUB_ETAG_TIMEOUT:=60}"
# 导出全部缓存路径相关环境变量，确保子进程和容器前检都可见
export TMPDIR VLLM_RPC_BASE_PATH
export TORCHINDUCTOR_CACHE_DIR TRITON_CACHE_DIR VLLM_CACHE_ROOT XDG_CACHE_HOME
export HF_HOME HF_DATASETS_CACHE HF_HUB_DOWNLOAD_TIMEOUT HF_HUB_ETAG_TIMEOUT
export TIKTOKEN_RS_CACHE_DIR
# 预创建所有缓存目录，任一步失败即中止环境初始化
mkdir -p "${TMPDIR}" \
    "${TORCHINDUCTOR_CACHE_DIR}" \
    "${TRITON_CACHE_DIR}" \
    "${VLLM_CACHE_ROOT}" \
    "${XDG_CACHE_HOME}" \
    "${HF_HOME}" \
    "${TIKTOKEN_RS_CACHE_DIR}" \
    "${HF_DATASETS_CACHE}" || return 1
```

.buildkite/scripts/ci-bake-rocm.sh

第三个提交加入的修复：validate_ci_build_context_source() 用 core.fileMode=false 忽略共享工作区可执行位漂移，并分段处理 git diff 退出码，避免 ROCm CI Docker context 校验报错，与 PR 的 CI 稳定性主题一致。

```
# 共享 AMD 工作区可能呈现可执行位漂移
# 上下文模式来自 pinned Git 树，因此只忽略文件系统模式变化
# 继续拒绝任何真实内容变化
if ! git -C "${source_root}" -c core.fileMode=false diff \
    --quiet --no-ext-diff --ignore-submodules=none --; then
    # git diff 退出码 1 表示有差异，>1 表示 git 本身执行失败
    git -C "${source_root}" -c core.fileMode=false diff \
        --quiet --no-ext-diff --ignore-submodules=none -- \
        || worktree_diff_status=$?
    if (( worktree_diff_status > 1 )); then
        printf 'Failed to inspect tracked CI worktree changes (git diff exited %s)\n' \
            "${worktree_diff_status}" >&2
        return 1
    fi
    echo "Tracked worktree changes cannot be omitted from the CI Docker context" >&2
    # 打印前 50 条差异，便于定位问题
    git -C "${source_root}" -c core.fileMode=false diff \
        --no-ext-diff --ignore-submodules=none --name-status -- \
        | sed -n '1,50p' >&2 || true
    git -C "${source_root}" -c core.fileMode=false diff \
        --no-ext-diff --ignore-submodules=none --summary -- \
        | sed -n '1,50p' >&2 || true
    return 1
fi
```

评论区精华

本 PR 没有实质性的代码讨论线程。claude[bot] 因 PR 来自 fork 而跳过自动 review（原话："This pull request is from a fork — automated review is disabled."），维护者 AndreasKaratzas 直接批准（"LGTM"）。技术判断集中在 PR body：作者明确指出故障根因是 job 级 **TMPDIR** 使 tiktoken vocab 缓存每次作废、请求时无重试下载，并提出 "只有第一个 job 下载 3.6 MB，之后全部读磁盘" 的收益；对缓存 key 随 URL 变化自动失效的设计也做了说明，无需手工清理。issue 评论中的 13 条消息全部是 **/ci run** 触发记录。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 缓存路径一致性：run-amd-test.sh 容器注入用 \${HF_MOUNT}/tiktoken-rs-cache，宿主机初始化用 \${HF_HOME}/...，若未来 HF_MOUNT 与 HF_HOME 映射错位，容器内会写错位置或重复下载；当前脚本两者对应同一持久卷。

- 并发首写竞态：多个 job 首次同时下载同一 vocab 文件到同一目录，若 openai-harmony 非原子写入可能产生半文件；属于低概率低影响。
- 文件模式校验放宽：ci-bake-rocm.sh 中 core.fileMode=false 会同时忽略真实的可执行位变更（例如新增脚本忘记 chmod +x），可能在 CI 构建上下文校验时漏检此类问题。
- 无自动化测试：改动全部在 CI 脚本，依赖手动验证；git diff 退出码分段逻辑（>1 视为失败）依赖 git 语义，未来 git 行为变化可能影响判定。
- 首次 job 仍依赖网络：缓存只在第一次 job 下载，若首次下载时网络故障，该 job 仍可能重演 500；但影响面从所有 job 缩小为单个 job。
- 影响：影响范围限定在 AMD/ROCm 硬件 CI (Buildkite amd-ci)，尤其 gpt-oss entryptoints 测试 (TestGPTOSSChat 8 例中 4 例曾失败)。改动后：后续 job 不再从 openaipublic.blob.core.windows.net 下载 vocab，3.6 MB 只被首次 job 拉取一次；消除了请求时下载 + 无重试带来的中途 500，显著降低测试 flake；ci-bake-rocm.sh 的校验放宽影响所有走 ROCm Docker 上下文构建的 job，能容忍共享工作区的模式伪差异；对用户服务与 vLLM 运行时零影响，纯 CI 基础设施变更。
- 风险标记：CI 脚本变更，缺少自动化测试，依赖环境变量约定，并发缓存首次写入

关联脉络

- PR #51721 [Bugfix][ROCm][CI] Stabilize build context and source caches: 与本 PR 同改 .buildkite/scripts/ci-bake-rocm.sh 与 ROCm Docker 构建流程，同属 AMD/ROCm CI 稳定性建设；一个是稳定构建上下文与源码缓存，一个是容忍共享工作区文件模式漂移。