

PR #56921 完整报告

vllm-project/vllm

[ROCm][CI] Prepare TheRock image for CI

合并时间: 2026-09-15 09:58

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56921>

执行摘要

- 一句话: 修复 ROCm 镜像 rocprofiler 双重注册并补齐 CI 冒烟阶段
- 推荐动作: 建议 CI/ 平台维护者精读本 PR: 核心亮点是用符号链接消除同一 inode 经双路径加载导致的动态加载器冲突, 这是一种值得沉淀的镜像构建手法。一般开发者可快速浏览, 无需深入。若后续 ROCm SDK 升级, 建议回归验证 rocprofiler 库清单是否需要扩展。

功能与动机

PR body 说明了两点动机: 一是 TheRock SDK 包把 rocprofiler 库以硬链接同时安装在 `_rocm_sdk_core` 和 `_rocm_sdk_devel`, 从两个路径加载导致 rocprofiler-register 在 `registration.cpp:233` 触发 CHECK-fail 并 SIGABRT, 杀掉 AMD CI 中每个启动 server 或 model-registry 子进程的 job; 二是 Dockerfile.rock 缺少 `test_smoke` / `export_test_smoke` 阶段, 导致 CI 的 smoke-test bake target (`smoke-test-rocm-ci`) 无法构建。改动目标是让 TheRock 镜像真正具备 CI 可用性。

实现拆解

1. 统一 rocprofiler 库规范路径 (`docker/Dockerfile.rock_base`): 在原有 ROCm SDK 库符号链接规范化逻辑 (将各库 `ln -sf` 到 `${ROCM_PATH}/lib`) 之后追加一个 for 循环, 针对 `librocprofiler-sdk.so.1`、`librocprofiler-register.so.0`、`librocprofiler-sdk-roctx.so.1` 三个库, 当 `_rocm_sdk_core/lib` 与 `_rocm_sdk_devel/lib` 同时存在时, 先 `rm -f` 删除 core 下的硬链接副本, 再 `ln -sf` 建立指向 devel 路径的符号链接, 最后执行 `ldconfig` 刷新动态链接器缓存。这样两条路径最终解析到同一 canonical 路径, 动态加载器按路径去重后只加载一次库, 避免 rocprofiler-register 注册两次触发 CHECK 失败。
2. 新增 CI 冒烟测试阶段 (`docker/Dockerfile.rock`): 新增 FROM test AS test_smoke 阶段, 接收 ROCM_SMOKE_ID 构建参数, 在 `--network=none` 下运行 `buildkite/scripts/rocm/smoke-test-image.sh --inside` 冒烟脚本, 成功后把 ROCM_SMOKE_ID (默认 local) 写入 `/vllm-smoke-ok` 标记文件; 随后新增 FROM scratch AS export_test_smoke 阶段, 通过 `COPY --from=test_smoke /vllm-smoke-ok /` 把标记文件拷入空镜像, 供 CI bake target 取回结果。
3. 验证: 改动随 AMD CI 在 TheRock 镜像上实测 (PR body 引用 Buildkite build 12914), 确认不再出现 rocprofiler 双重注册导致的测试失败, 且无新增回归; 随后合并了 main 分支保持镜像定义与主干同步。

关键文件：

- docker/Dockerfile.rock_base (模块 CI 镜像；类别 infra；类型 infrastructure)：核心修复所在：把 `_rocm_sdk_core` 下三个 `rocprofiler` 库从硬链接改为指向 `_rocm_sdk_devel` 的符号链接，从根源上消除双路径加载导致的 `rocprofiler-register CHECK-fail` 与 `SIGABRT`，是本次变更最关键的文件。
- docker/Dockerfile.rock (模块 CI 镜像；类别 infra；类型 infrastructure)：新增 `test_smoke` 与 `export_test_smoke` 两个构建阶段，使 CI 的 `smoke-test-rocm-ci bake target` 可以构建并取回冒烟结果，是本次 CI 准备工作的第二项关键变更。

关键符号：未识别

关键源码片段

docker/Dockerfile.rock_base

核心修复所在：把 `_rocm_sdk_core` 下三个 `rocprofiler` 库从硬链接改为指向 `_rocm_sdk_devel` 的符号链接，从根源上消除双路径加载导致的 `rocprofiler-register CHECK-fail` 与 `SIGABRT`，是本次变更最关键的文件。

```
# docker/Dockerfile.rock_base: ROCm SDK 库规范化（新增部分）
# 前置循环负责把各 rocm_sdk 库符号链接到 ROCM_PATH；新增循环再处理 rocprofiler 三库：
# SDK 包在 _rocm_sdk_core 与 _rocm_sdk_devel 安装了同一 inode 的硬链接，
# 经两条路径加载会触发 rocprofiler-register 双重注册 CHECK-fail (registration.cpp:233)
# 并 SIGABRT 进程，因此把 core 副本改写为指向 devel 的符号链接，统一为单一路径。
&& for f in librocprofiler-sdk.so.1 librocprofiler-register.so.0 \
    librocprofiler-sdk-roctx.so.1; do \
    if [ -e "${SITE_PACKAGES}/_rocm_sdk_core/lib/${f}" ] \
    && [ -e "${SITE_PACKAGES}/_rocm_sdk_devel/lib/${f}" ]; then \
    # 先删除硬链接副本，再建立符号链接，两步缺一不可 \
    rm -f "${SITE_PACKAGES}/_rocm_sdk_core/lib/${f}"; \
    ln -sf "${SITE_PACKAGES}/_rocm_sdk_devel/lib/${f}" \
        "${SITE_PACKAGES}/_rocm_sdk_core/lib/${f}"; \
    fi; \
done \
&& ldconfig
```

docker/Dockerfile.rock

新增 `test_smoke` 与 `export_test_smoke` 两个构建阶段，使 CI 的 `smoke-test-rocm-ci bake target` 可以构建并取回冒烟结果，是本次 CI 准备工作的第二项关键变更。

```
# docker/Dockerfile.rock: 新增 CI 冒烟测试阶段
# test_smoke 基于 test 镜像，在无网络环境下跑冒烟脚本并把结果写入标记文件；
# export_test_smoke 用 scratch 空镜像导出该标记，CI 的 smoke-test-rocm-ci
# bake target 由此可构建并校验冒烟结果。
FROM test AS test_smoke
ARG ROCM_SMOKE_ID
RUN --network=none \
    bash /vllm-workspace/.buildkite/scripts/rocm/smoke-test-image.sh --inside \
    && printf '%s\n' "${ROCM_SMOKE_ID:-local}" > /vllm-smoke-ok
```

```
FROM scratch AS export_test_smoke
COPY --from=test_smoke /vllm-smoke-ok /
```

评论区精华

PR 的 review 讨论非常少：claude[bot] 指出该 PR 来自 fork，自动审查被禁用，需维护者评论 @claude review 触发一次性审查；维护者 AndreasKaratzas 未触发额外审查，直接给出 APPROVED。没有出现关于技术方案的分歧或替代方案讨论，决策依据主要来自 PR body 中 AMD CI 的实测结果。

- Fork PR 自动审查被禁用 (other): 维护者未触发额外审查，AndreasKaratzas 直接批准合并，决策依据为 PR body 中 AMD CI 的实测结果。

风险与影响

- 风险：
 - 构建期非原子替换：rm -f 与 ln -sf 分两步执行，若镜像构建在两个命令之间中断，会留下缺失的库文件；不过 Docker 构建失败即中止该 layer，影响范围限于构建过程。
 - 库清单固定：只规范化了 librocprofiler-sdk.so.1、librocprofiler-register.so.0、librocprofiler-sdk-roctx.so.1 三个库名，若未来 SDK 版本新增其他双注册库，需要同步更新该列表，否则可能复发同类 SIGABRT。
 - 符号链接依赖 devel 组件：core 下的符号链接指向 _rocm_sdk_devel 路径，若后续镜像裁剪或调整 devel 组件，core 的符号链接会悬空，需要保持两个组件同时存在。
 - 冒烟脚本依赖：test_smoke 阶段硬编码依赖 .buildkite/scripts/rocm/smoke-test-image.sh 存在且可执行，脚本路径或行为变化会直接影响 bake target 构建。
 - 该变更属于镜像构建期修复，不触碰任何运行时代码，回归面限于 CI 镜像构建与冒烟流程，不涉及推理路径。
 - 影响：对 AMD CI 而言，本次变更修复了 TheRock 镜像上每个启动 server 或 model-registry 子进程的 job 都会遇到的 rocprofiler SIGABRT 崩溃，使该镜像可以实际投入 CI；同时补齐了 smoke-test bake target，CI 冒烟测试阶段可正常构建。影响范围集中在 docker 镜像定义与 CI 配置，不改变 vLLM 运行时行为，对用户和推理服务无直接影响。对团队的影响是 AMD CI 维护成本降低，TheRock 镜像从「不可用」变为「可上线」。
 - 风险标记：SIGABRT 级 CI 崩溃修复，镜像构建期非原子替换，依赖固定库清单与外部冒烟脚本

关联脉络

- PR #56783 [XPU][CI] update gpt-oss package version: 同为平台侧 CI/build 适配修复（XPU 的 gpt-oss 版本回退与安装修复），与本次 ROCm TheRock 镜像准备同属 AMD CI 平台保障工作线，虽是不同平台但目标一致。
- PR #56169 [CI] Check target branch freshness before starting CI: 同为 CI 基础设施改进，着眼点是 CI 触发前的分支新鲜度校验，与本 PR 的镜像准备互补，均服务于提升 CI 稳定性。