

PR #55246 完整报告

vllm-project/vllm

[ROCm][CI] Bump ROCK base to ROCm 10.0

合并时间: 2026-09-04 11:38

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/55246>

执行摘要

- 一句话: 升级 TheRock 基础镜像至 ROCm 10.0 并更新 CI 发布标签。
- 推荐动作: 这个 PR 值得精读, 特别是 `docker/Dockerfile.rock_base` 文件的变更。它清晰地展示了如何执行一次有计划、有范围控制的大型基础镜像升级: 精确锁定版本、简化构建流程、应用必要的运行时规避方案, 并同步更新 CI 配置。对于负责维护构建基础设施或计划进行类似平台迁移的工程师来说, 其中关于依赖管理 (如显式安装并验证 triton 版本)、清理过时步骤以及与相关 PR (#55014, #52826, #55099) 协同的做法具有很好的参考价值。

功能与动机

本 PR 旨在基于社区成员 @rasmith 提供的 `ROCm:vllm:rock10` dockerfile, 将 vllm 的 ROCK 基础镜像 (`docker/Dockerfile.rock_base`) 从 ROCm 7.14.0 升级到 10.0.0。这是 #55014 (构建并发布 ROCMk 夜间镜像) 的后续工作, 目的是更新基础工具链, 合并已确认的依赖升级 (如 AITER), 并规避已知的运行时问题 (如 torch 2.12 附带的 rocprofiler 队列互锁导致的 HSA 关闭挂起)。作者明确指出这是一个“第一次落地”的范围有限的变更, 许多更深入的改动 (如 Triton 源码重建) 将作为后续工作。

实现拆解

本次变更主要围绕基础设施和配置展开, 核心逻辑是更新基础镜像的组件版本并简化构建流程。

1. 核心版本更新: 在 `docker/Dockerfile.rock_base` 中, 将所有与 ROCm 版本相关的 ARG 从 7.14.0 更新为 10.0.0, 包括 `TORCH_VERSION`、`TORCHVISION_VERSION`、`TORCHAUDIO_VERSION` 和 `ROCM_SDK_VERSION`。同时, 将 `AITER_BRANCH` 从 `v0.1.19` 更新至 `v0.1.21.post1`, 以镜像已合入 main 的 #52826 变更。
2. Triton 安装方式简化: 移除了之前从源码构建 Triton 的 `build_triton` 阶段。改为通过新增的 `TRITON_VERSION` 参数 (值为 `3.8.0+git4cff872c.rocm10.0.0`) 在 `base` 阶段直接安装与 PyTorch 版本精确匹配的预构建 wheel。此简化消除了对 Triton 源码仓库和分支的依赖, 并删除了之前为覆盖 torch METADATA 而进行的复杂 sed 操作。
3. 清理过时步骤与注释: 根据 review 反馈, 删除了多个不再需要的构建步骤和注释, 包括: `base` 阶段冗余的 `python3 -c "import torch..."` 验证命令、AOTriton 构建后遗留的 `RUN ldconfig`、以及对 `TRITON_BRANCH/TRITON_REPO` 等已废弃参数的引用。这些清理减少了镜像层数和构建复杂性。

4. CI 管道标签更新：在 `.buildkite/release-pipeline.yaml` 中，将发布和清理夜间镜像的脚本调用参数从 `rocm714` 全部更改为 `rocm100`，确保 CI 流水线发布正确的新版本镜像。
5. 应用规避方案：在最终镜像阶段添加了环境变量 `ENV ROCPROFILER_QUEUE_INTERPOSITION=0`，以规避 `torch 2.12` 附带的 `rocprofiler-sdk` 在 HSA 关闭时的挂起问题，这与 `#55099` 中对 `Dockerfile.rocm_base` 的修复保持一致。

关键文件：

- `docker/Dockerfile.rock_base`（模块 构建镜像；类别 `infra`；类型 `infrastructure`；符号 `TRITON_VERSION`, `ROCM_SDK_VERSION`, `ROCPROFILER_QUEUE_INTERPOSITION`）：PR 的核心文件，包含了从 ROCm 7.14.0 到 10.0.0 的全部基础镜像构建变更，是理解本次升级具体影响的关键。
- `.buildkite/release-pipeline.yaml`（模块 CI 管道；类别 `config`；类型 `configuration`）：配套变更，将 CI 发布管道中的镜像标签从 `rocm714` 更新为 `rocm100`，确保新构建的镜像能被正确发布和清理。

关键符号：未识别

关键源码片段

`docker/Dockerfile.rock_base`

PR 的核心文件，包含了从 ROCm 7.14.0 到 10.0.0 的全部基础镜像构建变更，是理解本次升级具体影响的关键。

```
# 此变更展示了 Triton 安装流程的简化以及环境规避方案的引入。
# 旧版需要复杂的源码构建和 METADATA 修补，新版直接安装精确匹配的预构建 wheel。

# 在 base 阶段，将 triton 与 torch/torchvision/torchaudio 一同安装，确保版本完全匹配
ARG TRITON_VERSION=3.8.0+git4cff872c.rocm10.0.0
RUN pip install \
    --index-url ${ROCM_RELEASE_WHEELS_MULTIARCH_URL} \
    --extra-index-url ${ROCM_NIGHTLY_WHEELS_MULTIARCH_URL} \
    "torch==${TORCH_VERSION}" \
    "torchaudio==${TORCHAUDIO_VERSION}" \
    "torchvision==${TORCHVISION_VERSION}" \
    "triton==${TRITON_VERSION}"

# 在 final 阶段，直接安装所有 wheel 并验证 triton 版本，无需特殊处理
RUN --mount=type=bind,from=debs,src=/app/debs,target=/install \
    pip install /install/*.whl \
    && ! pip check 2>&1 | grep -iE '^(torch|triton) ' \
    && test "$(pip show triton | awk '/^Version:/ {print $2}')" = "${TRITON_VERSION}" \
# ... 后续创建约束文件等步骤 ...

# 应用已知问题的环境变量规避，以解决 torch 2.12 附带 rocprofiler 的队列挂起问题
ENV ROCPROFILER_QUEUE_INTERPOSITION=0
```

评论区精华

Review 讨论主要由作者 Rohan138 自己发起，集中于代码精简和必要规避方案的应用。

- 代码精简：作者在多条评论中要求移除自己在初始提交中添加但属于过渡性或冗余的注释与代码块（如对 TRITON_VERSION、whl-next URLs 的解释性注释，以及旧 Triton METADATA 修补逻辑），这反映了在追求清晰的同时保持 Dockerfile 简洁的意图。最终版本移除了这些内容，使核心构建逻辑更突出。
- 必要规避：关于添加 `ROCPROFILER_QUEUE_INTERPOSITION=0` 环境变量的决策，虽未在评论中详细讨论，但其提交信息和相关 PR (#55099) 表明这是一个已知的、必须应用的运行时兼容性修复。评审者 dllehr-amd 以“Let's make the bump”直接批准，表明对变更范围和必要性的认可。
- 清理 Dockerfile 中的冗余注释和代码 (design): 所有提出的清理项均被采纳，在最终提交中移除，使 Dockerfile 核心构建逻辑更清晰。
- 引入 `ROCPROFILER_QUEUE_INTERPOSITION=0` 环境变量 (correctness): 作为已知的、必要的运行时兼容性修复被接受并合并。

风险与影响

- 风险：主要风险在于基础镜像的重大版本跳跃可能引入的未预见兼容性问题，但本次变更已通过多种方式降低风险：
 1. 版本精确锁定：所有核心依赖（torch, triton, aiter）版本被精确锁定，避免了依赖解析带来的不确定性。
 2. 范围受限：变更为“第一次落地”，明确排除了大规模重构（如 Triton 源码重建），并只更新了基础镜像，未改动业务代码（Dockerfile.rock 未变）。
 3. 遵循已有修复：应用的 rocprofiler 规避方案已在其他镜像中验证。
 4. 已知限制：基于 commit 信息，作者已识别并计划跟进 TRITON_SOURCE 的重建、ROCm/CLB 的源码构建等未包含项，风险被主动管理。潜在的回归风险主要存在于依赖新版本 ROCm 10.0 的内核和运行时环境（如 AOTriton），但这属于平台升级的固有风险。
- 影响：对用户的影响较小，主要体现在开发者体验：
 - 正面影响：为使用 ROCm 平台的开发者提供了更新的、基于 ROCm 10.0 的官方基础镜像和夜间构建版本，可能包含新特性和性能改进。
 - 兼容性：作为基础镜像升级，依赖于此镜像的下游构建或工作流需重新构建以获取新版本。由于标签从 rocm714 变为 rocm100，外部引用旧标签的流水线将自动停止获取新镜像，避免意外破坏。

对系统的影响：

- 更新了 vllm CI/CD 基础设施中 ROCK 构建链的核心基础，简化了构建流程，减少了维护负担（如不再需要维护 Triton 源码构建）。

对团队的影响：

- 为后续基于 ROCm 10.0 的更深入优化（如启用新硬件特性）铺平了道路。
- 通过镜像 #52826 的 AITER 升级，确保了不同 Dockerfile（rocm_base 与 rock_base）间关键依赖的一致性。
- 风险标记：基础镜像升级，兼容性风险，范围受限的首次落地

关联脉络

- PR #55014 [ROCm][CI] Build and publish TheRock nightly docker images: 本 PR 是 #55014 的直接后续，完成了其构建的 ROCK 镜像的基础版本升级（从 ROCm 7.14 到 10.0）。
- PR #52826 [ROCm] Bump AITER to 0.1.21.post1: 本 PR 在 Dockerfile.rock_base 中镜像了 #52826 在 Dockerfile.rocm_base 中进行的 AITER 版本升级，确保不同构建变体依赖一致。
- PR #55099 [Dockerfile] Mitigate torch 2.12 rocprofiler hang: 本 PR 应用了与 #55099 相同的环境变量规避方案（ROCPROFILER_QUEUE_INTERPOSITION=0），以解决 torch 2.12 引入的运行时挂起问题。