

PR #41416 完整报告

vllm-project/vllm

[Build] Switch CUDA 13.0 wheel builds to PyTorch manylinux_2_28 base

合并时间: 2026-05-02 20:51

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/41416>

执行摘要

- 一句话: 切换 CUDA 13.0 wheel 构建至 manylinux_2_28 基础镜像以兼容旧系统
- 推荐动作: 值得精读。Dockerfile 中基于 BUILD_OS 的条件分支模式清晰展示了如何在同一构建流程中支持不同 OS 家族, 可作为后续迁移其他 CUDA 版本的参考。同时建议跟进 jikunshang 关于稳定镜像标签的提议以增强构建可复现性。

功能与动机

将 CUDA 13.0 wheel 的 glibc 下限从 2.34 降至 2.28, 匹配 PyTorch 官方 wheel, 解决 vllm wheel 在 RHEL 8、Ubuntu 20.04 等旧 OS 上无法使用的问题 (issue #26118)。同时为未来迁移至 manylinux 基础镜像奠定基础。

实现拆解

1. Dockerfile: 新增 BUILD_OS 构建参数 (默认 ubuntu), 在 base stage 根据其值分支: 若为 manylinux, 使用 dnf 安装系统依赖 (ccache、git、curl、sudo、rdma-core-devel) 并依赖镜像预装的 /opt/python/cpXY-cpXY/ Python 解释器; 若为 ubuntu, 沿用原 apt 安装方式并通过 uv 引导对应版本 Python。2..buildkite/release-pipeline.yaml: 将 x86_64 和 aarch64 的 CUDA 13.0 wheel 构建步骤的 BUILD_BASE_IMAGE 改为对应 manylinux 镜像, 并新增 BUILD_OS=manylinux 参数。
2. docker/versions.json: 注册 BUILD_OS 默认值。
3. 不变部分: CUDA 12.9、CPU、CUDA 13.0 的 runtime 镜像及其他构建保持不变, 确保改动影响范围最小。

关键文件:

- docker/Dockerfile (模块 构建镜像; 类别 infra; 类型 infrastructure): 核心变更, 新增 BUILD_OS 参数并实现 apt/dnf 分支安装逻辑, 控制构建基线和 Python 引导方式
- .buildkite/release-pipeline.yaml (模块 发布流水线; 类别 config; 类型 configuration): 配置 cu130 构建步骤使用 manylinux 镜像和 BUILD_OS 参数
- docker/versions.json (模块 镜像配置; 类别 infra; 类型 infrastructure): 注册 BUILD_OS 默认值, 保持与 Dockerfile 同步

关键符号: 未识别

关键源码片段

docker/Dockerfile

核心变更，新增 BUILD_OS 参数并实现 apt/dnf 分支安装逻辑，控制构建基线和 Python 引导方式

```
# 新增构建参数，默认使用 Ubuntu 路径
```

```
ARG BUILD_OS=ubuntu
```

```
# base stage: 根据 BUILD_OS 选择包管理器和 Python 引导方式
```

```
RUN if [ "${BUILD_OS}" = "manylinux" ]; then \
```

```
    # manylinux 路径：使用 dnf (AlmaLinux 8) 安装必要工具
```

```
    dnf install -y --setopt=install_weak_deps=False \
```

```
        ccache \
```

```
        git \
```

```
        curl \
```

```
        sudo \
```

```
        rdma-core-devel \
```

```
    && dnf clean all \
```

```
    && rm -rf /var/cache/dnf; \
```

```
else \
```

```
    # Ubuntu 路径：使用 apt 安装编译依赖并升级至 GCC 10
```

```
    apt-get update -y \
```

```
    && apt-get install -y --no-install-recommends \
```

```
        ccache \
```

```
        software-properties-common \
```

```
        git \
```

```
        curl \
```

```
        sudo \
```

```
        python3-pip \
```

```
        libibverbs-dev \
```

```
        gcc-10 \
```

```
        g++-10 \
```

```
    && update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-10 110 \
```

```
        --slave /usr/bin/g++ g++ /usr/bin/g++-10 \
```

```
    && (apt-get install -y --no-install-recommends python${PYTHON_VERSION}-dev 2>/dev/null
```

```
    || true) \
```

```
    && rm -rf /var/lib/apt/lists/* \
```

```
    # 使用 uv 引导目标 Python 版本并创建虚拟环境
```

```
    && curl -LsSf https://astral.sh/uv/install.sh | sh \
```

```
    && $HOME/.local/bin/uv venv /opt/venv --python ${PYTHON_VERSION}; \
```

```
fi
```

```
ENV PATH=/opt/venv/bin:$PATH
```

评论区精华

review 中有两处核心讨论：

1. tlrnchlsmith 指出 Dockerfile 中一处针对 Ubuntu 20.04 的 python-dev 安装逻辑在切换到 22.04 后已不再必要，建议移除；mgoin 表示将在单独 PR 处理。

2. jikunshang 提议对 manylinux 基础镜像使用固定版本标签（如 cuda13.0-v2.11.0-rc6）以避免 PyTorch 侧更新导致意外回归，该建议暂未得到回复，属于未解决的潜在风险。
- 移除 Ubuntu 20.04 遗留的 python-dev 安装 (style): 将于后续 PR 移除，当前 PR 暂未处理。
 - 建议使用固定版本的 manylinux 镜像标签 (design): 暂未处理，风险已记录。

风险与影响

- 风险：主要风险包括：
 1. manylinux 镜像的 Python 解释器版本与 uv 引导的版本可能不同，但已统一使用 /opt/python/cpXY-cpXY/ 路径，风险较低；
 2. dnf 包名与 apt 不完全一致（如 rdma-core-devel 替代 libibverbs-dev），已验证可通过；
 3. manylinux 镜像标签为动态 :cuda13.0，PyTorch 侧更新可能带入不兼容变动，需关注镜像发布时间；
 4. 双分支构建路径（apt/dnf）增加了 Dockerfile 维护成本，但当前改动覆盖全面。
- 影响：影响范围限于 CUDA 13.0 wheel 的构建过程和产物：
 - 用户：cu130 wheel 现在可广泛应用于 RHEL 8 (glibc 2.28) 及以上系统，扩大用户覆盖。
 - 系统：构建镜像大小可能与之前不同，但无运行时影响。
 - 团队：需维护两种系统包管理器路径，并关注 manylinux 镜像的更新节奏。
 - 风险标记：动态基础镜像标签，双包管理器维护

关联脉络

- 暂无明显关联 PR