

PR #6466 完整报告

verl-project/verl

[ci] chore: update npu docker to cann 9.0.0

合并时间: 2026-05-26 09:25

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6466>

执行摘要

- 一句话: 更新 Ascend NPU Docker 至 CANN 9.0.0
- 推荐动作: 建议仔细阅读 review 中的建议, 并在后续 PR 中修复遗留问题。PR 本身完成了版本升级的目标, 但实施细节有瑕疵。对于使用 Ascend NPU 的开发者, 应关注修复后的 Dockerfile 版本。

功能与动机

升级 CANN 基础版本至 9.0.0, 以支持最新的 Ascend 硬件和软件特性, 保持 NPU 环境的兼容性。

实现拆解

1. 新增 Dockerfile: 在 docker/ascend/ 下新增 Dockerfile.ascend_9.0.0_a2 和 Dockerfile.ascend_9.0.0_a3, 基于 CANN 9.0.0 镜像, 安装 vLLM 0.18.0、vLLM-ascend、MindSpeed、Megatron-LM 等依赖。
2. 更新 CI 工作流: 修改 .github/workflows/docker-build-ascend-a2.yml 和 a3.yml, 将引用的 Dockerfile 路径和镜像标签从 8.5.0 改为 9.0.0。
3. 更新文档: 在 docs/ascend_tutorial/get_start/dockerfile_build_guidance.rst 中添加 CANN 9.0.0 的镜像表格, 并修正版本号。
4. 修改依赖文件: 从 requirements-npu.txt 中移除 triton-ascend==3.2.0 (review 建议应更新至 3.2.1)。

关键文件:

- docker/ascend/Dockerfile.ascend_9.0.0_a2 (模块 Docker 镜像; 类别 infra; 类型 infrastructure) : 核心新增文件, 定义 CANN 9.0.0 环境
- docker/ascend/Dockerfile.ascend_9.0.0_a3 (模块 Docker 镜像; 类别 infra; 类型 infrastructure) : 类似 A2 但针对 A3 设备
- .github/workflows/docker-build-ascend-a2.yml (模块 CI 配置; 类别 infra; 类型 infrastructure) : CI 工作流引用新 Dockerfile
- .github/workflows/docker-build-ascend-a3.yml (模块 CI 配置; 类别 infra; 类型 infrastructure) : 类似 A2 的 CI 变更

- docs/ascend_tutorial/get_start/dockerfile_build_guidance.rst（模块 文档说明；类别 docs；类型 documentation）：更新文档以反映新版本
- requirements-npu.txt（模块 依赖配置；类别 docs；类型 documentation）：移除了 triton-ascend 依赖

关键符号：未识别

关键源码片段

[docker/ascend/Dockerfile.ascend_9.0.0_a2](#)

核心新增文件，定义 CANN 9.0.0 环境

```
# 基于 CANN 9.0.0 基础镜像
```

```
FROM swr.cn-south-1.myhuaweicloud.com/ascendhub/cann:9.0.0-910b-ubuntu22.04-py3.11
```

```
ARG SOC_VERSION="ascend910b1"
```

```
# 安装系统依赖
```

```
RUN apt-get update -y && \
```

```
    apt-get install -y --no-install-recommends gcc g++ cmake libnuma-dev wget git curl jq vim
```

```
    build-essential && \
```

```
    apt-get clean && \
```

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

```
    pip install --upgrade pip packaging setuptools==80.10.2 && \
```

```
    pip cache purge
```

```
# 克隆第三方库
```

```
RUN ARCH=$(uname -m) && \
```

```
    echo "[LOG INFO] Detected architecture: $ARCH" && \
```

```
    if [ "$ARCH" = "x86_64" ]; then \
```

```
        pip config set global.extra-index-url "https://download.pytorch.org/whl/cpu/"; \
```

```
    fi && \
```

```
    git clone --depth 1 --branch v0.18.0 https://github.com/vllm-project/vllm.git && \
```

```
    git clone -b releases/v0.18.0 https://github.com/vllm-project/vllm-ascend.git && \
```

```
    git clone https://gitcode.com/Ascend/MindSpeed.git && \
```

```
    cd MindSpeed && git checkout 2.3.0_core_r0.12.1 && cd .. && \
```

```
    git clone --depth 1 --branch core_v0.12.1 https://github.com/NVIDIA/Megatron-LM.git
```

```
# 安装第三方库（注意：此处使用 export 而非 ENV，环境变量不会持久化到后续 RUN）
```

```
RUN export PIP_EXTRA_INDEX_URL=https://triton-ascend.osinfra.cn/pypi/simple/ && \
```

```
    export PIP_TRUSTED_HOST=triton-ascend.osinfra.cn && \
```

```
    ARCH=$(uname -m) && \
```

```
    if [ "$ARCH" = "aarch64" ]; then \
```

```
        export LD_LIBRARY_PATH=/usr/local/Ascend/cann-9.0.0/aarch64-linux/devlib/linux/
```

```
        aarch64:$LD_LIBRARY_PATH; \
```

```
    elif [ "$ARCH" = "x86_64" ]; then \
```

```
        export LD_LIBRARY_PATH=/usr/local/Ascend/cann-9.0.0/x86_64-linux/devlib/linux/x86_64/:
```

```
        $LD_LIBRARY_PATH; \
```

```
    fi && \
```

```
# source 命令在 dash shell 中不可用，应使用 . 代替
source /usr/local/Ascend/ascend-toolkit/set_env.sh && \
source /usr/local/Ascend/nna/ab/set_env.sh && \
cd vllm && pip install -r requirements/build.txt && \
VLLM_TARGET_DEVICE=empty pip install -v -e . && cd .. && \
cd vllm-ascend && pip install -r requirements.txt && \
export COMPILE_CUSTOM_KERNELS=1 && pip install -v -e . && cd .. && \
pip install -e MindSpeed && \
pip install -e Megatron-LM && \
pip uninstall -y triton && \
pip install mbridge && \
rm -rf /tmp/* /var/tmp/* && \
pip cache purge
```

```
# 为 DeepSeek 模型修改 mbridge: 将 cuda 替换为 npu (使用硬编码行号，脆弱)
RUN MBRIDGE_PATH=$(pip show mbridge | grep Location | awk '{print $2}') && \
TARGET_FILE="${MBRIDGE_PATH}/mbridge/models/ext/deepseek_v3/dequant_fp8_safetensor_
io.py" && \
sed -i '34s/cuda/npu/;51s/cuda/npu/' "$TARGET_FILE"
```

```
# 克隆并安装 verl (使用 git clone 而非 COPY, 无法包含本地变更)
RUN git clone --recursive https://github.com/verl-project/verl.git && \
cd verl && pip install -r requirements-npu.txt && pip install -v -e . && cd .. && \
rm -rf /tmp/* /var/tmp/* && \
pip cache purge
```

评论区精华

Review 由 gemini-code-assist[bot] 提出多个高优先级问题:

- 在 RUN 中使用 export 设置环境变量不会持久化，后续 RUN 无法使用，应改用 ENV 指令。
- source 命令在 Docker 默认 /bin/sh (dash) 中不可用，应使用 . 命令。
- sed 使用硬编码行号 (34s,51s) 修改 mbridge 文件，脆性大，应改用模式匹配。
- Dockerfile 中通过 git clone 获取 verl 代码，导致无法包含 PR 本地变更，应使用 COPY 指令。
- 文档中 A2/A3 9.0.0 的链接错误指向了 8.5.0 文件。
- requirements-npu.txt 中直接删除 triton-ascend 可能破坏非 Docker 安装，应更新版本号。
这些意见均未在合并前得到修复或回应，但 PR 仍被 wucong25 批准合并。
- Dockerfile 中 export 应改为 ENV (correctness): 建议但未采纳，PR 已合并。
- source 命令在 dash 中不可用 (correctness): 建议但未采纳。
- sed 硬编码行号脆弱 (design): 建议但未采纳。
- git clone 应改为 COPY (design): 建议但未采纳。
- 文档 URL 错误 (documentation): 建议但未采纳。
- 移除 triton-ascend 应改为更新版本 (correctness): 建议但未采纳。

风险与影响

- 风险：主要风险来自 review 中指出的 Dockerfile 缺陷：
 1. 使用 export 而非 ENV 导致后续 RUN 命令无法获取环境变量，可能造成 pip 安装失败。
 2. source 命令在 dash 下不可用，构建时可能报错。
 3. sed 硬编码行号在 mbridge 更新后可能失效。
 4. 使用 git clone 而非 COPY 导致 CI 无法验证本地变更，也影响本地开发。
 5. 删除 triton-ascend 可能导致依赖缺失，影响用户环境。这些风险可能导致 NPU Docker 构建不稳定。
- 影响：影响 Ascend NPU 环境的容器构建流程和 CI 流水线。用户若手动构建或使用 CI 构建，可能遇到构建失败。文档更新帮助用户了解新版本，但部分链接错误。团队需尽快修复 review 中指出的问题以确保构建可靠性。
- 风险标记：Dockerfile 构建可能失败，环境变量不持久，source 命令兼容性，sed 硬编码脆弱，依赖移除风险

关联脉络

- PR #6816 [ci] fix: fix Megatron-Bridge version in e2e_ppo_trainer_megatron_sglang_ascend.yml and update megatron for sglang ascend: 同样涉及 Ascend Dockerfile 和 CI 配置修改，属于 NPU 基础设施持续改进的一部分。
- PR #6711 [ci] chore: solve daplo error and add three baselines for npu nightly ci: 也修改了 NPU CI 工作流和依赖，与本 PR 的 NPU Docker 升级同属 Ascend 环境维护线。