

PR #38371 完整报告

vllm-project/vllm

Enable building MoRI with AMD AINIC stack

合并时间: 2026-04-21 02:18

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/38371>

执行摘要

- 一句话: ROCm Docker 构建新增 MoRI NIC 后端支持
- 推荐动作:
 - 对于使用 AMD NIC 的 ROCm 用户, 此 PR 提供了清晰的构建方式, 值得采用。
 - 对于 vLLM 基础镜像维护者, 值得关注这种多阶段构建分离模式, 在延长底层镜像生命周期和增加应用层灵活性之间做了良好平衡。
 - 建议后续添加对其他 NIC 后端 (如 Broadcom BNXT) 的支持时采用类似模式。

功能与动机

升级 MoRI 到 v1.1.0 以支持运行时自动检测 NIC 后端, 并允许用户在构建时通过 `NIC_BACKEND` 参数 (如 `ainic`) 安装对应 NIC 驱动包, 满足 AMD Pensando NIC 部署需求。参考 SGLang 的 Docker 实现, 保持构建灵活性和基础镜像稳定性。

实现拆解

1. 在 `Dockerfile.roc_base` 中, 将 `MORI_BRANCH` 从固定 SHA 升级到 `v1.1.0`, 移除之前的 `USE_IONIC` 等编译标志, 因为新版 MoRI 通过运行时环境变量自动检测 NIC。
2. 在 `Dockerfile.roc` 顶部新增构建参数 `NIC_BACKEND` (默认 `none`)、`AINIC_VERSION` 和 `UBUNTU_CODENAME`。
3. 新增 `mori_base` 构建阶段, 安装 MoriIO 代理 Python 依赖 (`blinker`, `quart`, `msgpack`, `aihttp`, `pyzmq`), 并通过 `case` 语句根据 `NIC_BACKEND` 值执行不同操作:
 - `none`: 跳过 NIC 驱动安装。
 - `ainic`: 添加 AMD AINIC 仓库, 安装 `libionic-dev` 和 `ionic-common`。
 - 其他值: 报错退出。
4. 将 `test` 和 `final` 阶段的基础镜像从 `base` 改为 `mori_base`, 确保这两个阶段包含 MORI 运行时和依赖。
5. 在最终镜像中, 向 `/app/versions.txt` 追加 `MORI_NIC_BACKEND` 和 `AINIC_VERSION` 信息。

测试: 通过 `docker build` 测试默认和 `ainic` 两种场景, 验证构建成功和版本文件内容。

关键文件:

- docker/Dockerfile.rocm (模块 构建脚本; 类别 infra; 类型 infrastructure) : 核心变更文件: 新增 `mori_base` 构建阶段, 通过 `NIC_BACKEND` 参数条件安装 NIC 驱动和代理依赖; 将 `test` 和 `final` 阶段基础镜像改为 `mori_base`, 确保运行时包含 MORI。
- docker/Dockerfile.rocm_base (模块 构建脚本; 类别 infra; 类型 configuration) : 升级 MORI 分支到 v1.1.0, 移除了之前用于指定 NIC 的编译标志 (`USE_IONIC` 等), 依赖运行时自动检测。这是基础镜像的唯一变更, 影响所有 ROCm 镜像。

关键符号: 未识别

关键源码片段

docker/Dockerfile.rocm

核心变更文件: 新增 `mori_base` 构建阶段, 通过 `NIC_BACKEND` 参数条件安装 NIC 驱动和代理依赖; 将 `test` 和 `final` 阶段基础镜像改为 `mori_base`, 确保运行时包含 MORI。

```
FROM base AS mori_base
ARG NIC_BACKEND
ARG AINIC_VERSION
ARG UBUNTU_CODENAME
```

```
RUN /bin/bash -lc 'set -euo pipefail; \
  echo "[MORI] Install MoRI proxy deps"; \
  # 安装 MoriIO 代理所需 Python 包, 用于分离式 / KV  workflow
  pip install --quiet --ignore-installed blinker && \
  pip install --quiet quart msgpack aiohttp pyzmq; \
  echo "[MORI] NIC_BACKEND=${NIC_BACKEND}"; \
  case "${NIC_BACKEND}" in \
    none) \
      # 默认: 不安装 NIC 驱动, 使用 mlx5
      ;; \
    ainic) \
      # AMD AINIC: 添加官方仓库并安装驱动与开发包
      apt-get update && apt-get install -y --no-install-recommends \
        ca-certificates curl gnupg apt-transport-https && \
      rm -rf /var/lib/apt/lists/* && mkdir -p /etc/apt/keyrings; \
      curl -fsSL https://repo.radeon.com/rocm/rocm.gpg.key | gpg --dearmor > /etc/apt/keyrings/
      amdainic.gpg; \
      echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/amdainic.gpg] https://repo.radeon.com/
      amdainic/pensando/ubuntu/${AINIC_VERSION} ${UBUNTU_CODENAME} main" \
        > /etc/apt/sources.list.d/amdainic.list; \
      apt-get update && apt-get install -y --no-install-recommends \
        libionic-dev \
        ionic-common \
      ; \
      rm -rf /var/lib/apt/lists/*; \
      ;; \
    *) \
      echo "ERROR: unknown NIC_BACKEND=${NIC_BACKEND}. Use one of: none, ainic"; \
```

```
    exit 2; \  
    ;; \  
esac;'
```

```
# 后续 test 和 final 阶段从 mori_base 继承，确保运行时包含 MORI  
FROM mori_base AS test  
...  
FROM mori_base AS final
```

评论区精华

- [tjtanaa](#) 提出将 NIC 后端构建从基础镜像移到应用镜像层，避免用户因 NIC 后端选择而必须重建耗时的基础镜像（PyTorch、aiter 等）。[ichbinblau](#) 最初尝试在 `Dockerfile.rocm_base` 中添加逻辑，后根据建议迁移至 `Dockerfile.rocm`。
- [gshtras](#) 强烈反对在 `Dockerfile.rocm` 中添加额外构建步骤，认为应保持简洁，建议要么在基础镜像中默认构建，要么支持从预构建制品安装。
- [billishyahao](#) 指出 MoRI v1.1.0 已支持运行时 NIC 自动检测，仅需升级版本并在 `Dockerfile.rocm` 中安装用户态库。
- 最终达成一致：基础镜像只升级 MoRI 版本（无 NIC 编译标志），在 `Dockerfile.rocm` 中新增 `mori_base` 阶段处理 NIC 依赖，基础镜像无需重建。
- [gemini-code-assist\[bot\]](#) 指出 `apt-get` 调用效率问题（多次 `update`）和 `shell` 变量使用不当（`$$` 应改为 `$`），这些在最终版本中已修正。
- NIC 后端构建放在哪个 `Dockerfile` 层 (design): 最终在 `Dockerfile.rocm` 中新增 `mori_base` 阶段处理 NIC 依赖；基础镜像仅升级 MoRI 版本。
- `apt-get` 操作效率 (performance): 已修复：最终版本中在 `mori_base` 阶段内只执行一次 `update` 和 `clean`。
- `Shell` 变量语法错误 (style): 已修复：修改为 `$VAR`。
- MoRI v1.1.0 NIC 自动检测 (design): 采纳：移除编译标志，仅升级版本；NIC 用户态包安装留在应用层 `Dockerfile`。

风险与影响

- 风险：
 - 外部依赖风险：AINIC 仓库地址可能变化或不可达，导致构建失败。
 - 构建效率：新增 `mori_base` 阶段增加约 2 分钟构建时间（用于安装代理依赖和 NIC 包）。
 - 兼容性：默认 `NIC_BACKEND=none` 行为不变，但 `final` 和 `test` 阶段从 `base` 改为 `mori_base` 可能带入意外依赖（如 `pyzmq`），但仅影响 ROCm 镜像。
 - 安全性：安装自第三方仓库的包（AMD AINIC）可能引入信任问题，但已验证位于官方 `repo.radeon.com`。
- 影响：
 - 用户：ROCm Docker 用户现在可以通过 `--build-arg NIC_BACKEND=ainic` 构建支持 AMD Pensando NIC 的镜像；默认构建不变。

- 系统：镜像大小增加约（NIC 驱动和 Python 依赖），具体取决于选择的后端。
- 团队：维护者需保持 NIC 仓库 URL 和版本号更新；CI 中可能需要添加对不同 NIC 后端的构建验证。
- 风险标记：外部仓库依赖，构建时间增加 2 分钟，基础镜像更换影响运行时，条件构建复杂度

关联脉络

- 暂无明显关联 PR