

PR #36434 完整报告

sgl-project/sglang

[AMD] Add ROCm 10 (gfx942 / gfx950) release images

合并时间: 2026-08-28 23:03

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36434>

执行摘要

- 一句话: 新增 ROCm 10 发布镜像与 CI 流程, 支持 gfx942/gfx950
- 推荐动作: 值得关注 Dockerfile 中 rocm1000-base 阶段的 wheel-based 栈组装方式, 以及发布工作流中按架构隔离 device wheels 的设计。这是基础设施演进, 对 AMD 社区用户有实际价值, 但不涉及核心算法, 精读优先级中等。

功能与动机

支持新一代 AMD 硬件 MI350/MI355X (gfx950) 以及 MI300/MI325 (gfx942), 并切换到 AMD 官方稳定 wheel 通道获取 ROCm SDK, 避免依赖 apt 基础镜像。PR body 明确说明使用 AMD's stable wheel-based ROCm SDK, 固定了完整软件栈版本, 确保镜像可复现。

实现拆解

1. 新增 rocm1000-base 构建阶段 (docker/rocm.Dockerfile): 从 ubuntu:24.04 基础镜像安装 ROCm 10 SDK 与 PyTorch wheels, 而不是使用发布好的基础镜像。SDK 落入 site-packages, 因此需要做路径修复; 同时为 gfx942-rocm1000 和 gfx950-rocm1000 新增构建目标, 每个镜像只安装目标架构的 device wheels, 避免跨架构包污染。
2. 新增发布工作流 (.github/workflows/release-docker-amd-rocm10.yml): 支持每日 cron 和手动触发, 按 GPU 架构矩阵构建, 并发布到 rocm/sgl-dev 与 lmsysorg/sglang-rocm, 镜像 tag 形如 v{version}-rocm10-mi30x-{date}。
3. 扩展 AMD CI 依赖安装脚本 (scripts/ci/amd/amd_ci_install_dependency.sh): 识别 -rocm1000 镜像后缀, 选择正确的 extras (dev_hip_rocm724), 并在构建 MORI 时安装 libgrpc++-dev, 处理 noble 基础镜像缺失依赖的问题。
4. 扩展测试工作流选项: 在 nightly-test-amd-rocm720.yml、pr-test-amd-extra.yml、pr-test-amd-rocm720.yml 中增加 rocm10 选项, 允许显式选择 ROCm 10 镜像测试; 同时移除性能步骤名称中硬编码的 ROCm 版本, 避免后续维护时误导。

关键文件:

- docker/rocm.Dockerfile (模块 镜像构建; 类别 infra; 类型 infrastructure; 符号 rocm1000-base, gfx942-rocm1000, gfx950-rocm1000): 核心变更文件, 新增 rocm1000-base 阶段和 gfx942/gfx950 的 ROCm 10 构建目标, 定义了 wheel-based 的 ROCm 10 SDK 组装方式及路径修复逻辑。

- `.github/workflows/release-docker-amd-rocm10.yml` (模块 发布流程; 类别 infra; 类型 infrastructure) : 新增的发布工作流, 定义了每日 / 手动触发、构建矩阵、镜像标签和推送目标, 是 ROCm 10 镜像对外发布的入口。
- `scripts/ci/amd/amd_ci_install_dependency.sh` (模块 CI 脚本; 类别 infra; 类型 infrastructure) : 修改 AMD CI 依赖安装逻辑, 新增 rocm1000 镜像后缀识别与 extras 选择, 确保 ROCm 10 镜像在 CI 中依赖正确。
- `.github/workflows/nightly-test-amd-rocm720.yml` (模块 测试流程; 类别 infra; 类型 infrastructure) : 修改 nightly 测试工作流, 增加 rocm10 选项并移除性能步骤名称中的硬编码 ROCm 版本, 使工作流可复用。
- `.github/workflows/pr-test-amd-extra.yml` (模块 测试流程; 类别 infra; 类型 infrastructure) : 为 AMD extra 测试工作流增加 rocm10 选择项, 便于 PR 阶段针对性验证。
- `.github/workflows/pr-test-amd-rocm720.yml` (模块 测试流程; 类别 infra; 类型 infrastructure) : 为 AMD PR 测试工作流增加 rocm10 选择项, 支撑 ROCm 10 镜像的 PR 验证。

关键符号: 未识别

关键源码片段

`docker/rocm.Dockerfile`

核心变更文件, 新增 rocm1000-base 阶段和 gfx942/gfx950 的 ROCm 10 构建目标, 定义了 wheel-based 的 ROCm 10 SDK 组装方式及路径修复逻辑。

```
# GPU_ARCH=-rocm1000 的目标使用 AMD stable wheel 通道组装 ROCm 10 栈,
# 而不是基于 apt 的 ROCm 基础镜像; 参见下方 rocm1000-base 阶段。
ARG BASE_IMAGE_ROCM1000="ubuntu:24.04"
# 默认从 rocm1000-base 阶段构建, 也可预构建后通过这两个参数覆盖。
ARG BASE_IMAGE_942_ROCM1000="rocm1000-base"
ARG BASE_IMAGE_950_ROCM1000="rocm1000-base"
```

```
# =====
# Shared ROCm 10.0.0 base for gfx942 and gfx950.
# 从纯净的 Ubuntu 24.04 基础镜像开始, 依次安装 AMD stable wheels,
# 使每个输出镜像只携带自身 GPU 架构的 device payload。
# SDK 被安装到 site-packages 而不是 /opt/rocm, 因此后续 Dockerfile
# 与 AITER 需要做路径修复才能找到 ROCm 组件。
# 选择 Python 3.12 是因为 3.13/3.14 缺少 st_attn、vsa、petit_kernel、
# wave-lang 等依赖的 wheel 包, pip 无法解析。
FROM $BASE_IMAGE_ROCM1000 AS rocm1000-base
```

```
# 在此阶段重新声明 GPU 架构选择器, 使矩阵构建可以按架构分支安装
# 对应的 ROCm device wheels。
```

`.github/workflows/release-docker-amd-rocm10.yml`

新增的发布工作流，定义了每日 / 手动触发、构建矩阵、镜像标签和推送目标，是 ROCm 10 镜像对外发布的入口。

```
name: Release Docker Images Nightly ROCm 10 (AMD)
on:
  workflow_dispatch:
  inputs:
    job_select:
      description: 'Select which release job to run'
      type: choice
      default: 'all'
      options: ['all', 'publish']
    gpu_arch:
      description: 'Select which ROCm 10 GPU arch to build'
      type: choice
      default: 'all'
      options: ['all', 'gfx942-rocm1000', 'gfx950-rocm1000']
  schedule:
    - cron: '0 12 * * *'

jobs:
  publish:
    strategy:
      fail-fast: false
      matrix:
        gpu_arch: ['gfx942-rocm1000', 'gfx950-rocm1000']
    steps:
      # 构建并推送镜像，根据 GPU_ARCH 映射到 rocm10-mi30x / rocm10-mi35x tag
      # 镜像最终同时发布到 rocm/sgl-dev 与 lmsysorg/sglang-rocm
```

scripts/ci/amd/amd_ci_install_dependency.sh

修改 AMD CI 依赖安装逻辑，新增 rocm1000 镜像后缀识别与 extras 选择，确保 ROCm 10 镜像在 CI 中依赖正确。

```
# 根据镜像内的 GPU_ARCH 环境变量判断基础镜像后缀，决定使用哪个依赖 extra。
if [[ "${IMAGE_GPU_ARCH}" =~ ^(gfx942|gfx950)(-rocm720|-rocm724|-rocm1000)?$ ]]; then
  case "${IMAGE_GPU_ARCH}" in
    *-rocm724) IMAGE_BASE_ARG_SUFFIX="_ROCM724"; IMAGE_STAGE_SUFFIX="-rocm724" ;;
    *-rocm720) IMAGE_BASE_ARG_SUFFIX="_ROCM720"; IMAGE_STAGE_SUFFIX="-rocm720" ;;
    *-rocm1000) IMAGE_BASE_ARG_SUFFIX="_ROCM1000"; IMAGE_STAGE_SUFFIX="-rocm1000" ;;
    *) IMAGE_BASE_ARG_SUFFIX=""; IMAGE_STAGE_SUFFIX="" ;;
  esac
fi

# ROCm 7.2.4 与 ROCm 10 镜像都携带 torch 2.11，而 srt_hip 依赖的
# compressed-tensors 0.15.0 要求 torch<2.11，所以统一选择 dev_hip_rocm724 extras。
if [[ "${IMAGE_STAGE_SUFFIX}" == "-rocm724" || "${IMAGE_STAGE_SUFFIX}" == "-rocm1000" ]]; then
```

```
EXTRAS="${EXTRAS}/dev_hip/dev_hip_rocm724}"  
fi
```

评论区精华

Review 中无评论，仅有两位 reviewer 的 APPROVE (yctseng0211 和 HaiShaw)。作者在 issue 评论中记录了四轮验证：ROCm RC4 与 GA 版本分别搭配两个 AITER 提交（默认 pin 与 a6d2b564），每次均跑完整的 AMD PR/nightly 套件，验证覆盖了 MI300/MI35x 的构建与测试。

- 暂无高价值评论线程

风险与影响

- 风险：

1. SDK 路径调整风险 (docker/rocm.Dockerfile)：ROCm SDK 安装在 site-packages 而非 /opt/rocm，可能影响 AITER、MORI 等依赖路径的组件，需要后续维护者持续关注。
2. 外部 wheel 通道依赖：构建依赖 AMD stable repo 的可用性与 CDN 稳定性，若上游 wheel 被移除或变更，镜像构建可能失败。
3. 发布权限与安全：新增发布工作流使用 environment: 'prod'，涉及 Docker Hub secrets，需要确保权限最小化。
4. CI 回归：测试矩阵新增 rocm10 选项，但未修改默认的全行为，已通过完整验证，风险相对可控。
 - 影响：对 AMD 用户，可获取基于 ROCm 10 的官方镜像，支持新一代 GPU；对 CI 维护者，测试矩阵新增一个选项，发布流程新增每日任务；对发布团队，需维护新的镜像 tag 约定。整体影响范围集中在 AMD 平台的基础设施层面，不改动核心推理逻辑。
 - 风险标记：新硬件支持，外部依赖通道，CI 流程扩展，基础设施变更

关联脉络

- PR #35319 关注中：AMD ROCm 基础设施相关 PR: PR body 中直接引用了该 PR，可能是 ROCm 构建或镜像相关的早期工作。
- PR #36379 fix(lora): build the MoE LoRA align JIT kernel on ROCm: 同为 AMD 平台基础设施改动，共享 CI 与镜像构建链路，可能依赖本 PR 引入的 ROCm 10 镜像。
- PR #36758 [AMD] Qwen3.5 ASM FMHA chunked-prefill context attention: AMD 性能优化，需依赖 ROCm 镜像版本，与 ROCm 10 发布镜像形成配套。
- PR #35341 [AMD][Fix] Qwen3.5: make empty-batch guard tuple-aware on fused AR+quant path: AMD 平台 bugfix，与本 PR 同属 AMD 生态演进，关注镜像兼容性。