

PR #26466 完整报告

sgl-project/sglang

[NPU] chore: basic software upgrade

合并时间: 2026-05-29 10:04

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26466>

执行摘要

- 一句话: NPU 平台基础软件升级 (CANN 9.0.0、PyTorch 2.10.0)
- 推荐动作: 此 PR 为 NPU 平台的必需基础设施升级, 建议尽快合并并密切观察后续 CI 和 nightly 测试结果。值得关注的设计决策是 set-image-config job 的使用, 它为多镜像环境下的版本管理提供了可扩展模式。建议在后续 PR 中跟进解决以下问题: 将 nightly 测试的重复代码提取为公共脚本; 将 Dockerfile 中硬编码版本替换为变量引用; 与 sgl-kernel-npu 团队协调发布修复版本。

功能与动机

NPU 平台需要定期升级依赖以保持与最新 CANN (9.0.0) 和 PyTorch (2.10.0) 的兼容性, 同时修复已知问题 (如 sgl-kernel-npu 中的 Qwen3.5 崩溃)。Issue 评论中要求参考 PR#20860 更新文档, 说明文档同步也是动机之一。

实现拆解

实现分为几个主要部分:

1. 核心依赖版本升级: 修改 scripts/ci/npu/npu_ci_install_dependency.sh 和 docker/npu.Dockerfile, 将所有相关依赖更新至新版本: 移除 OPTIONAL_DEPS 分支统一使用 PyTorch 2.10.0; 更新 memfabric-hybrid 至 1.0.8 并新增 memfabric-zbal 1.1.1; 将 triton-ascend 从 PyPI 包改为固定 URL 安装 (版本 3.2.1); 更新 sgl-kernel-npu 的 tag 至 2026.05.01.post3; 同步 Dockerfile 中的 ARG 变量确保默认版本一致。
2. CI 镜像配置重构: 在 .github/workflows/pr-test-npu.yml 中新增 set-image-config job, 将 CANN 镜像标签作为该 job 的输出, 其他 test job 通过 needs 依赖获取镜像配置, 避免硬编码。
3. 同步更新其他 CI 和发布工作流: 更新 .github/workflows/nightly-test-npu.yml、release-docker-npu.yml、release-docker-npu-nightly.yml 中的 CANN 版本 (8.5.0 → 9.0.0) 和 sgl-kernel-npu tag, 保持全局一致性; nightly 测试中的 torchaudio 版本同步升级到 2.10.0。

关键文件:

- scripts/ci/npu/npu_ci_install_dependency.sh (模块 CI 脚本; 类别 infra; 类型 infrastructure): 核心安装脚本, 集中管理所有 NPU 依赖的版本和安装方式, PR 中变更最大, 直接体现软件升级内容。

- docker/npu.Dockerfile (模块 Docker 构建; 类别 infra; 类型 infrastructure) : Docker 构建文件, 定义了镜像内的基础软件版本, 与安装脚本协同确保环境一致性。
- .github/workflows/pr-test-npu.yml (模块 CI 工作流; 类别 infra; 类型 infrastructure) : PR 测试工作流, 重构了镜像配置管理, 通过 set-image-config job 集中管理 CANN 镜像标签, 是 infrastructure 层面的重要改进。
- .github/workflows/nightly-test-npu.yml (模块 CI 配置; 类别 infra; 类型 infrastructure) : 夜间测试工作流, 同步更新了 CANN 镜像和 torchaudio 版本, 保持与主 CI 一致。
- .github/workflows/release-docker-npu.yml (模块 发布配置; 类别 infra; 类型 infrastructure) : NPU Docker 发布工作流, 更新了 CANN 版本和 sgl-kernel-npu tag。
- .github/workflows/release-docker-npu-nightly.yml (模块 发布配置; 类别 infra; 类型 infrastructure) : NPU 夜间 Docker 发布工作流, 同步更新 CANN 版本和 sgl-kernel-npu tag。

关键符号: 未识别

关键源码片段

scripts/ci/npu/npu_ci_install_dependency.sh

核心安装脚本, 集中管理所有 NPU 依赖的版本和安装方式, PR 中变更最大, 直接体现软件升级内容。

```
### Install MemFabric (分布式共享内存库)
${UV_PIP_INSTALL} memfabric-hybrid==1.0.8

### Install PyTorch and PTA (PyTorch 和 NPU 适配层)
# 统一使用 PyTorch 2.10.0, 不再区分 OPTIONAL_DEPS 分支
PYTORCH_VERSION="2.10.0"
TORCHVISION_VERSION="0.25.0"
TORCHAUDIO_VERSION="2.10.0"
${UV_PIP_INSTALL} torch==${PYTORCH_VERSION} torchvision==${TORCHVISION_VERSION}
torchaudio==${TORCHAUDIO_VERSION} \
  --index-url ${TORCH_CACHE_URL:="https://download.pytorch.org/whl/cpu"} \
  --extra-index-url ${PYPI_CACHE_URL:="https://pypi.org/simple/"}
# PTA 需要从 GitCode 下载 (UV 不支持, 故使用 PIP)
PTA_URL="https://gitcode.com/Ascend/pytorch/releases/download/v26.0.0-pytorch2.10.0/torch_
npu-2.10.0-cp311-cp311-manylinux_2_28_aarch64.whl"
${PIP_INSTALL} ${PTA_URL}

### Install zbal (负载均衡库, 仅 NPU 需要)
${UV_PIP_INSTALL} memfabric-zbal==1.1.1

### Install Triton-Ascend (NPU 上的 Triton 后端, 指定版本 3.2.1)
${PIP_INSTALL} "https://gitcode.com/Ascend/triton-ascend/releases/download/v3.2.1/triton_
ascend-3.2.1-cp311-cp311-manylinux_2_27_aarch64.manynlinux_2_28_aarch64.whl"

### Install sgl-kernel-npu (自定义 NPU 算子集合)
SGLANG_KERNEL_NPU_TAG="2026.05.01.post3"
```

```
# 根据设备类型下载对应架构的预编译包
(cd sgl-kernel-npu && wget ...) # 省略具体下载解压命令
```

docker/npu.Dockerfile

Docker 构建文件，定义了镜像内的基础软件版本，与安装脚本协同确保环境一致性。

```
# 基础软件版本定义（可构建时覆盖）
ARG PYTORCH_VERSION="2.10.0"
ARG TORCHVISION_VERSION="0.25.0"
ARG TORCHAUDIO_VERSION="2.10.0"
# NPU 适配层 PTA 的下载 URL（区分 arm64 和 amd64）
ARG PTA_URL_ARM64="https://gitcode.com/Ascend/pytorch/releases/download/v26.0.0-
pytorch2.10.0/torch_npu-2.10.0-cp311-cp311-manylinux_2_28_aarch64.whl"
ARG PTA_URL_AMD64="https://gitcode.com/Ascend/pytorch/releases/download/v26.0.0-
pytorch2.10.0/torch_npu-2.10.0-cp311-cp311-manylinux_2_28_x86_64.whl"
# Triton-Ascend 下载 URL（同样区分架构）
ARG TRITON_URL_ARM64="https://gitcode.com/Ascend/triton-ascend/releases/download/v3.2.1/
triton_ascend-3.2.1-cp311-cp311-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl"
ARG TRITON_URL_AMD64="https://gitcode.com/Ascend/triton-ascend/releases/download/v3.2.1/
triton_ascend-3.2.1-cp311-cp311-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl"
```

评论区精华

Review 中主要讨论了以下问题：

- 版本不匹配问题：gemini-code-assist 指出 Dockerfile 中默认 PYTORCH_VERSION 仍为 2.8.0，但 PTA URL 已指向 2.10.0，存在不一致风险。该问题已在后续 commit 中修复。
- 硬编码建议：建议在 sgl-kernel-npu 的下载命令中使用 \${PYTORCH_VERSION} 变量代替硬编码 torch2.10.0。该建议未被采纳，当前仍为硬编码。
- OPTIONAL_DEPS 移除：Makcum888e 和 ping1jing2 指出当 OPTIONAL_DEPS 非空时需要安装 PyTorch 2.10.0，最终 PR 直接移除了该条件分支，统一升级。
- 重复代码提取：ping1jing2 建议将 nightly-test-npu.yml 中多处重复的安装命令提取为公共脚本放入 scripts/ci/npu/，以减少维护成本。该建议未被实现，仍保留重复。
- CI 镜像管理方案：ping1jing2 建议使用 env 变量来定义镜像标签，但 PR 最终采用了新增 set-image-config job 的方式。两种方案各有利弊，但 set-image-config 在并行 trigger 场景下更灵活。
- sgl-kernel-npu 已知 bug：VDV1985 指出升级后的 tag 2026.05.01.post3 仍包含 Qwen3.5 的崩溃 bug，需要创建新的 release。PR 后可能需要进一步跟进。
 - Dockerfile 版本不匹配 (correctness): 已修复，最终版本中 PYTORCH_VERSION 已更新为 2.10.0。
 - sgl-kernel-npu 下载使用变量 (design): 未采纳，仍保留硬编码。
 - OPTIONAL_DEPS 分支清理 (refactor): 已解决，统一使用 PyTorch 2.10.0。
 - nightly 测试重复代码抽象 (testing): 未实现，仍保留重复。
 - CI 镜像标签管理方式 (design): 采用 set-image-config job，该方案在并行触发时更灵活。

- sgl-kernel-npu 已知崩溃 bug (correctness): 待跟进, PR 已使用 post3 tag, 但需要后续发布修复版本。

风险与影响

- 风险: 主要风险包括:
 - 兼容性风险: 新版本的 CANN 9.0.0、PyTorch 2.10.0、PTA 等可能与现有模型代码存在兼容性问题, 需通过 CI 验证。
 - 已知组件 bug: sgl-kernel-npu 2026.05.01.post3 被指出包含 Qwen3.5 崩溃的 bug, 可能导致特定模型测试失败。
 - CI 配置变更风险: set-image-config job 的引入改变了 job 间的依赖关系, 可能影响 CI 触发逻辑 (如 skip 条件)。需要确认 if 条件是否正确, 避免测试被意外跳过。
 - 硬编码不一致风险: Dockerfile 中 sgl-kernel-npu 的下载路径仍硬编码了 torch2.10.0, 若未来 PyTorch 版本再次变更, 容易遗漏更新。
- 影响: 此 PR 主要影响 NPU 后端的开发者和 CI 维护者:
 - 开发者: 升级后的基础软件可能带来性能提升或新功能, 但也可能引入临时兼容性问题。需要关注 CI 反馈。
 - 系统: NPU 的 Docker 镜像和 CI 将使用新版本依赖, 确保与官方支持一致。
 - 团队: CI 维护者需要适应新的镜像管理方式 (set-image-config), 减少未来升级时的工作量。但 nightly 测试中的重复代码问题未被解决, 仍存在维护负担。
 - 风险标记: 依赖版本冲突风险, 已知组件 bug 未修复, CI 配置变更影响范围广, 硬编码版本维护隐患

关联脉络

- 暂无明显关联 PR