

# PR #24801 完整报告

sgl-project/sglang

[CI][NPU] use internal HTTP cache for Rust toolchain via RUSTUP\_CACHE

合并时间: 2026-05-13 14:58

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24801>

## 执行摘要

- 一句话: NPU CI 启用内部 HTTP 缓存加速 Rust 工具链安装
- 推荐动作: 值得读者关注 `install_rustup.sh` 中的条件分支模式, 它展示了如何优雅地支持缓存优先安装策略, 并注意 `GITHUB_PATH` 这类 CI 环境的边界问题。

## 功能与动机

作为 #23514 的后续, NPU CI 所在的 aarch64 runner 依赖 Rust 工具链, 原有 `rsproxy.cn` 镜像虽快但不在团队直接控制下。集群内已有用于 PyPI 等的 `nginx` 缓存, 将其用于 `rustup` 可复用基础设施、减少外部依赖并加速安装。

## 实现拆解

1. 修改 CI 工作流 (`pr-test-npu.yml`): 在所有 NPU job 中, 将原先分别设置的 `RUSTUP_DIST_SERVER` 和 `RUSTUP_UPDATE_ROOT` 替换为一个统一的环境变量 `RUSTUP_CACHE_URL=http://cache-service.nginx-pypi-cache.svc.cluster.local:8082`。
2. 重写安装脚本 (`install_rustup.sh`): 脚本先导出 `CARGO_HOME/bin` 到 `GITHUB_PATH` 以保证后续步骤能找到 Rust 二进制。随后根据是否设置 `RUSTUP_CACHE_URL` 决定安装方式: 若设定了, 则从缓存 URL 构建 Rustup Dist Server, 手动下载对应架构的 `rustup-init` 并执行安装; 否则回退到官方 `curl` 安装方式。两个分支均移除了 `--default-toolchain stable` 标志以避免不一致。

关键文件:

- `.github/workflows/pr-test-npu.yml` (模块 CI 工作流; 类别 `infra`; 类型 `infrastructure`): 核心 CI 配置文件, 定义了所有 NPU job 的环境变量, 将外部 `rustup` 源切换为内部缓存。
- `scripts/ci/utils/install_rustup.sh` (模块 安装脚本; 类别 `infra`; 类型 `infrastructure`): 安装脚本, 包含条件分支实现缓存优先安装, 并修复了 `GITHUB_PATH` 导出问题。

关键符号: 未识别

## 关键源码片段

`scripts/ci/utils/install_rustup.sh`

安装脚本, 包含条件分支实现缓存优先安装, 并修复了 `GITHUB_PATH` 导出问题。

```
# 确保 cargo/rustc 在后续步骤中可见
```

```

export PATH="${CARGO_HOME:-$HOME/.cargo}/bin:${PATH}"
if [ -n "${GITHUB_PATH:-}" ]; then
  echo "${CARGO_HOME:-$HOME/.cargo}/bin" >> "${GITHUB_PATH}"
fi

# 如果设置了 RUSTUP_CACHE_URL (NPU CI 使用内部缓存) , 则从缓存下载
if [ -n "${RUSTUP_CACHE_URL:-}" ]; then
  export RUSTUP_DIST_SERVER="${RUSTUP_CACHE_URL}/rustup"
  export RUSTUP_UPDATE_ROOT="${RUSTUP_CACHE_URL}/rustup/rustup"
  # 检测架构并下载对应的 rustup-init
  case "$(uname -m)" in
    x86_64) RUSTUP_ARCH="x86_64-unknown-linux-gnu" ;;
    aarch64) RUSTUP_ARCH="aarch64-unknown-linux-gnu" ;;
    *) echo "ERROR: unsupported arch $(uname -m)"; exit 1 ;;
  esac
  RUSTUP_TMP="$(mktemp -d)"
  trap 'rm -rf "${RUSTUP_TMP}"' EXIT
  curl --retry 3 --retry-delay 2 -sSfL \
    "${RUSTUP_UPDATE_ROOT}/dist/${RUSTUP_ARCH}/rustup-init" \
    -o "${RUSTUP_TMP}/rustup-init"
  chmod +x "${RUSTUP_TMP}/rustup-init"
  "${RUSTUP_TMP}/rustup-init" -y --no-modify-path
else
  # 回退到官方安装脚本
  curl --proto '=https' --tls1.2 --retry 3 --retry-delay 2 -sSf https://sh.rustup.rs \
    | sh -s -- -y --no-modify-path
fi

```

## 评论区精华

gemini-code-assist[bot] 指出原版改动移除了 `GITHUB_PATH` 更新逻辑, 会导致后续 GitHub Actions 步骤找不到 `cargo/rustc`, 另外缓存分支额外加了 `--default-toolchain stable` 造成不一致。Goalina 随后修复: 将 `GITHUB_PATH` 导出移到脚本顶部, 并统一移除 `--default-toolchain` 标志。

- `GITHUB_PATH` 回归与 `--default-toolchain` 一致性 (correctness): 修复完成, 两个分支现在行为一致且 `GITHUB_PATH` 正确设置。

## 风险与影响

- 风险: 变更仅影响 NPU CI 环境中的 Rust 安装路径。主要风险是新的内部缓存服务不可用时安装可能失败, 但脚本保留了未设置 `RUSTUP_CACHE_URL` 时的 fallback 至官方源。已在 review 中修复了 `GITHUB_PATH` 回归问题, 当前版本无已知回归。
- 影响: 仅对 NPU CI 的 Rust 工具链安装环节生效, 安装速度预期更快且更稳定 (依赖内部基础设施而非公共镜像)。不影响其他 CI 或用户功能。
- 风险标记: 仅影响 NPU CI 环境, 回归风险已修复

## 关联脉络

- PR #23514 [CI][NPU] Set RUSTUP\_DIST\_SERVER and RUSTUP\_UPDATE\_ROOT to rsproxy.cn: 本 PR 是 #23514 的后续, 将之前的 rustup 镜像切换为内部缓存。