

# PR #23017 完整报告

sgl-project/sglang

ci: install rust toolchain in ci\_install\_dependency.sh

合并时间: 2026-04-17 14:18

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/23017>

## 执行摘要

- 一句话: 在 CI 依赖安装脚本中新增 Rust 工具链安装步骤, 确保源码构建时 cargo/rustc 可用。
- 推荐动作: 该 PR 变更直接且必要, 适合快速浏览以理解 CI 依赖管理策略。值得关注的设计决策是复用 install\_protoc.sh 的模式, 将工具链安装封装为可重用脚本, 体现了基础设施代码的模块化思想。

## 功能与动机

PR body 明确指出, 由于 sglang wheel 通过 setuptools-rust 捆绑了原生 Rust gRPC crate (rust/sglang-grpc/), 任何源码安装 (包括 CI 执行的可编辑安装) 都会在构建时调用 cargo/rustc。如果 PATH 中缺少 Rust 编译器, 安装会失败并报错“error: can't find Rust compiler”。这是一个兜底修复, 确保无论 CI 运行器镜像是否预装 Rust 工具链, 都能正常工作。

## 实现拆解

1. 新增 Rust 安装脚本: 创建 scripts/ci/utils/install\_rustup.sh, 该脚本首先检查 cargo 和 rustc 是否已安装, 若已存在则直接退出; 否则通过 curl 从 sh.rustup.rs 安装 rustup, 并更新 PATH 确保后续步骤可用。
2. 集成到 CUDA CI 依赖脚本: 修改 scripts/ci/cuda/ci\_install\_dependency.sh, 在安装 protoc 后调用 install\_rustup.sh, 并更新步骤完成标记。
3. 集成到 NPU CI 依赖脚本: 修改 scripts/ci/npu/npu\_ci\_install\_dependency.sh, 在环境变量设置后调用 install\_rustup.sh, 并更新 PATH。
4. 配套调整: 无测试或文档配套改动, 纯基础设施脚本调整。

关键文件:

- scripts/ci/utils/install\_rustup.sh (模块 CI 脚本; 类别 infra; 类型 infrastructure): 新增的核心脚本, 负责检测并安装 Rust 工具链, 是本次变更的基石。
- scripts/ci/cuda/ci\_install\_dependency.sh (模块 CI 脚本; 类别 infra; 类型 infrastructure): CUDA 平台的主要 CI 依赖安装脚本, 集成了 Rust 安装步骤, 影响大部分 CI 流水线。
- scripts/ci/npu/npu\_ci\_install\_dependency.sh (模块 CI 脚本; 类别 infra; 类型 infrastructure): NPU 平台的 CI 依赖安装脚本, 同样集成了 Rust 安装, 确保 Ascend

NPU CI 也能正常工作。

关键符号：未识别

## 关键源码片段

### scripts/ci/utils/install\_rustup.sh

新增的核心脚本，负责检测并安装 Rust 工具链，是本次变更的基石。

```
#!/bin/bash
# 确保为源码构建的crate（如通过setuptools-rust捆绑的原生gRPC扩展）安装Rust工具链（rustc/
cargo）。
# 最小支持版本为1.85（2024版）。
set -euxo pipefail

# 如果之前CI步骤已安装rustup，则拾取cargo。
export PATH="${CARGO_HOME:-$HOME/.cargo}/bin:${PATH}"

if command -v cargo >/dev/null 2>&1 && command -v rustc >/dev/null 2>&1; then
    echo "rust already installed: $(rustc --version), $(cargo --version)"
    exit 0 # 已安装则提前退出，避免重复安装
fi

echo "rust not found, installing via rustup..."

# rustup.rs需要curl——确保其存在。
if ! command -v curl >/dev/null 2>&1; then
    if command -v apt-get &> /dev/null; then
        apt-get update || true
        apt-get install -y --no-install-recommends curl ca-certificates
    elif command -v yum &> /dev/null; then
        yum install -y curl ca-certificates
    else
        echo "ERROR: curl is required to install rustup, but no supported package manager was
found"
        exit 1 # 无支持的包管理器时失败
    fi
fi

# 通过官方脚本安装rustup，使用非交互模式且不修改PATH（后续手动处理）。
curl --proto '=https' --tlsv1.2 --retry 3 --retry-delay 2 -sSf https://sh.rustup.rs \
    | sh -s -- -y --no-modify-path

# 使cargo/rustc对当前shell和后续GitHub Actions步骤可见。
export PATH="${CARGO_HOME:-$HOME/.cargo}/bin:${PATH}"
if [ -n "${GITHUB_PATH:-}" ]; then
    echo "${CARGO_HOME:-$HOME/.cargo}/bin" >> "${GITHUB_PATH}" # 更新GitHub
Actions的PATH
fi
```

```
# 验证安装成功。
rustc --version
cargo --version
```

## 评论区精华

无 review 评论，PR body 中作者说明了变更原因和测试计划，并提及了相关 PR #23014 和 #23016（它们也做了类似改动，但可后续合并）。

- 暂无高价值评论线程

## 风险与影响

- 风险：低风险。主要风险点：
- 脚本健壮性：install\_rustup.sh 依赖 curl 和包管理器（apt-get/yum），若运行在不支持的环境中可能失败；但脚本已包含错误处理。
- 环境污染：自动安装 Rust 可能覆盖或干扰现有工具链，但脚本在检测到已安装时会提前退出，降低了冲突风险。
- 依赖版本：脚本未指定 Rust 版本，默认安装最新稳定版，可能与项目要求的特定版本不匹配；但 PR body 提到最小支持版本为 1.85，需确保兼容性。
- 影响：影响范围有限但关键：
- 对系统：仅影响 CI 构建流程，确保源码安装成功，避免因缺失 Rust 导致的构建中断。
- 对用户：无直接影响，属于后台基础设施改进。
- 对团队：提高了 CI 的可靠性和可重复性，减少了因环境差异导致的构建失败。
- 风险标记：环境依赖风险

## 关联脉络

- PR #23014 [main image] also install rust toolchain: PR body 中提及，同样在主镜像中安装 Rust 工具链，属于同一批 CI 改进。
- PR #23016 [release images] also install rust toolchain: PR body 中提及，同样在发布镜像中安装 Rust 工具链，属于同一批 CI 改进。