

PR #44981 完整报告

vllm-project/vllm

[Rust Frontend] [CI] Unify Rust artifact builds with setuptools-rust

合并时间: 2026-06-10 17:48

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/44981>

执行摘要

- 一句话: 统一 Rust 构件构建入口为 setuptools-rust
- 推荐动作: 建议关注其“单一入口”设计模式——未来扩展 Rust 模块只需在 tools/build_rust.py 中添加 RustExtension 条目, 其余环节自动适配。值得精读 tools/build_rust.py 和 setup.py 的改动以理解共享构建的原理。

功能与动机

原先存在两套 Rust 构建路径: CI 上手动调用 cargo, 本地通过 setup.py 触发 setuptools-rust。一旦需要添加多个 Rust 扩展模块, 平台相关的后缀、ABI 标签和安装路径必须由 setuptools-rust 统一处理, 而不是在 shell 或 Docker 中重复。此外, BuildKit 缓存 /workspace/rust/target 偶尔损坏导致构建失败, 必须移除。PR body 原文: "it becomes fragile once we introduce more Rust extension modules" 和 "Also removed BuildKit cache on /workspace/rust/target as it seems to use corrupted cache sometimes"。

实现拆解

1. 新增共享构建入口 tools/build_rust.py: 定义 rust_extensions() 函数 (返回 RustExtension 列表, 接收 optional 参数) 和 build_binary() 函数 (内嵌 setup 调用, 指定 build_rust 命令)。该文件成为 Rust 构件的唯一声明点。
2. 修改 setup.py: 移除 import setuptools_rust.Binding 和 RustExtension 的直接使用; 通过 load_module_from_path 动态加载 tools/build_rust.py 模块并调用其 rust_extensions(), 将结果传给 setup() 的 rust_extensions 参数。这样保持 setup.py 结构的同时复用共享定义。
3. 简化 build_rust.sh: 脚本不再指定 VLLM_RS_TARGET_PATH, 转而依赖 setuptools-rust 的 --inplace 输出路径。
4. 修改各 Dockerfile (Dockerfile、Dockerfile.cpu、Dockerfile.nightly_torch、Dockerfile.xpu、Dockerfile.rocm): 将 rust-build 阶段的基础镜像从 BUILD_BASE_IMAGE 改为 base, 确保 Python 环境可用; 安装 protoc、复制 tools/build_rust.py 和 build_rust.sh; 移除手动安装 rustup 的步骤 (改用 requirements/build/rust.txt 中的 Python 包依赖); 移除 BuildKit target 缓存挂载; 构建后从预设包路径复制构件 (vllm/vllm-rs 而非自定义路径)。

5. 新增 `requirements/build/rust.txt`: 锁定 `setuptools`、`setuptools-rust` 及 `wheel` 版本, 确保构建环境一致。
6. 更新 `MANIFEST.in`: 包含 `tools/build_rust.py`, 保证 `sdist` 中携带该文件。
7. 修正 `rust/README.md`: 反映新的构建路径。

关键文件:

- `tools/build_rust.py` (模块 构建脚本; 类别 `source`; 类型 `core-logic`; 符号 `rust_extensions`, `build_binary`, `main`): 核心新增文件, 定义了共享的 `RustExtension` 声明和构建入口, 是统一构建路径的基石。
- `setup.py` (模块 安装配置; 类别 `source`; 类型 `dependency-wiring`): 通过导入使用 `tools/build_rust.py` 替代直接定义 `RustExtension`, 体现共享入口的集成。
- `docker/Dockerfile` (模块 Docker 镜像; 类别 `infra`; 类型 `infrastructure`): Docker 构建阶段调整为使用 `base` 镜像, 复制 `build_rust.py` 并移除 `target` 缓存, 体现统一构建路径的部署落地。其他平台 `Dockerfile` 同步修改。
- `build_rust.sh` (模块 构建脚本; 类别 `other`; 类型 `core-logic`): 简化了脚本, 不再指定 `VLLM_RS_TARGET_PATH`, 与共享入口对接。
- `requirements/build/rust.txt` (模块 依赖配置; 类别 `docs`; 类型 `documentation`): 新增依赖文件, 锁定 `setuptools-rust` 和 `wheel` 版本, 确保构建环境一致。

关键符号: `rust_extensions`, `build_binary`, `main`

关键源码片段

`setup.py`

通过导入使用 `tools/build_rust.py` 替代直接定义 `RustExtension`, 体现共享入口的集成。

```
# 在函数 load_module_from_path 之后, 新增加载 tools/build_rust.py
envs = load_module_from_path("envs", os.path.join(ROOT_DIR, "vllm", "envs.py"))
# 新增: 加载共享构建模块
rust_build = load_module_from_path(
    "rust_build", os.path.join(ROOT_DIR, "tools", "build_rust.py")
)

VLLM_TARGET_DEVICE = envs.VLLM_TARGET_DEVICE
# ... 其他代码不变 ...

# 设置 rust_extensions 时调用共享函数
rust_extensions = rust_build.rust_extensions(
    optional=not should_require_rust_frontend()
)

setup(
    # ... 其他参数 ...
    rust_extensions=rust_extensions,
    # ...
)
```

评论区精华

自动化安全工具 [depthfirst-app\[bot\]](#) 提出了两个关于依赖版本的安全评论：

- requirements/build/rust.txt 中 `setuptools>=77.0.3` 受 CVE-2025-47273 影响（路径遍历），建议提升至 `>=78.1.1`。
- `wheel` 未固定版本，建议 `>=0.46.2` 避免 CVE-2026-24049。这些评论未在 PR 中被回复或采纳。此外，评审者 [Harry-Chen](#) 批准了 PR 并触发了 release pipeline 验证，但因主分支其他 bug 失败。[njhill](#) 口头认可设计。
- setuptools 版本存在 CVE (security): 未在 PR 中处理，未回复或采纳，PR 已合并。
- wheel 版本未固定 (security): 未处理。

风险与影响

- 风险：
 1. 构建缓存移除风险：移除了 `/workspace/rust/target` 的缓存挂载，尽管解决了偶发损坏问题，但每次有 Rust 变更时 Docker 构建时间可能增加 <5 分钟。
 2. 基础镜像变更风险：rust-build 阶段改用 base 镜像，可能引入 Python 版本兼容问题（已在后续修复 commit 中处理）。
 3. 新依赖风险：新增 requirements/build/rust.txt 引入 setuptools-rust 版本约束，未来升级可能需同步。
 4. 安全风险：未采纳的安全建议可能遗留潜在 CVE，但影响范围有限（wheel 仅用于构建时，且 setuptools 版本仍在 77-81 之间）。- 影响：对用户：无直接影响，Rust 前端二进制正常打包。对开发者：统一了本地和 CI 的 Rust 构建逻辑，降低了认知负担。对 CI/CD：Docker 构建步骤整合，减少维护点；但移除 target 缓存可能导致增量构建时间增加。对团队：为后续添加新 Rust 扩展模块（如推理或 tokenizer 扩展）提供了清晰的注册入口，预计 PR #44624 会直接受益。
- 风险标记：BuildKit 缓存移除，CVE 未修复，基础镜像变更

关联脉络

- PR #40848 [Rust Frontend] Integrate Rust frontend build system: PR #40848 最初引入了 Rust 前端的集成和独立 CI 构建阶段，本 PR 在此之上统一构建路径。
- PR #44624 Add more Rust extension modules (mentioned in body): PR #44624 将引入更多 Rust 扩展模块，本 PR 的统一构建入口为其奠定基础。