

PR #34864 完整报告

sgl-project/sglang

[CI] Path-gate Rust workspace tests in lint

合并时间: 2026-08-15 13:41

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34864>

执行摘要

- 一句话: Rust 工作区测试按路径门控, 仅相关变更时运行
- 推荐动作: 值得快速阅读, 是一个标准且可复用的路径门控模式, 适合对其他 CI 工作流参考。尤其值得注意的是将 workflow 自身纳入过滤集, 使门控变更也能触发完整验证。没有深度技术权衡, 不必精读。

功能与动机

PR body 明确指出: PR #34860 仅修改文档, 但其 lint 任务在精确恢复 390 MB Rust 缓存后仍花费 1m44s 执行 `cargo test --workspace`, 其中测试仅运行 0.47s, `test-profile` 编译占用了几乎全部时间。仅文档变更和无关 Python 变更不会影响 Rust 工作区, 因此应跳过该步骤以节省 CI 时间。

实现拆解

整个变更集中在 `.github/workflows/lint.yml`, 实现步骤如下:

1. 新增“Detect Rust workspace changes”步骤, 使用 `dorny/paths-filter@v3` 定义 `rust_workspace` 过滤器, 覆盖 `rust/**`、`proto/sglang/runtime/v1/sglang.proto` (`rust/sglang-grpc/build.rs` 编译所依赖的 proto 来源) 以及 `.github/workflows/lint.yml` 自身。
2. 为已有的“Run rust/ workspace tests”步骤添加 `if: steps.paths.outputs.rust_workspace == 'true'` 条件, 保留原有 Rust 缓存与 `timeout 900` 配置不变。
3. 校验配套: 作者在本 PR 上运行了 `pre-commit` (`SKIP=no-commit-to-branch`)、对路径过滤器、条件、超时和 Cargo 命令做了 YAML 结构断言, 并扫描了 Cargo 路径依赖与 `build-script` 输入, 确保路径集合完整。
4. 影响: 合并后, 文档或无关 Python 变更的 lint 运行将完全跳过 1m45s 的 Cargo 测试步骤; 而改动 `lint.yml` 本身或 Rust/proto 相关文件的 PR 仍会执行该步骤, 形成自举保证。

关键文件:

- `.github/workflows/lint.yml` (模块 CI 工作流; 类别 `infra`; 类型 `ci-gating`): 唯一的变更文件, 引入 Rust workspace 测试的路径门控, 减少无关变更的 CI 耗时。

关键符号: 未识别

评论区精华

该 PR 没有实质性的讨论评论，审阅者 hnyls2002 直接批准且未留下文字意见。PR 作者在评论区三次使用 /rerun-failed-ci，表明曾重跑失败 CI，但均未附带额外讨论。整体是一个简洁、无争议的基础设施改动。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于路径门控的完整性：若未来 Rust 工作区新增依赖或构建输入文件，而该路径列表未同步更新，可能导致 Rust 变更未被测试而悄悄跳过。此外 dorny/paths-filter@v3 使用可变 tag 而非固定 SHA，存在第三方 Action 被替换的供应链风险；proto 路径 proto/sglang/runtime/v1/sglang.proto 仅覆盖单一文件，后续 proto 结构变化需要维护。整体风险较低，不会影响运行时行为。
- 影响：影响范围主要在 CI/lint 工作流：对仅文档或无关 Python 变更的 PR，lint 任务将节省约 1m45s 的 Rust 测试时间（该步骤占比最大），提升开发者反馈速度；对 Rust、proto、lint workflow 自身的变更，行为保持不变，仍会执行完整 Rust workspace 测试。该改动不影响任何运行时逻辑、模型行为或 API 兼容性。
- 风险标记：路径门控可能漏检，第三方 Action 版本未锁定，依赖路径列表维护

关联脉络

- 暂无明显关联 PR