

PR #37618 完整报告

sgl-project/sglang

[CI] Add /rerun-test --changed to rerun every test file a PR modifies

合并时间: 2026-09-05 13:38

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/37618>

执行摘要

- 一句话: 为 /rerun-test 命令新增 --changed 标志, 自动重运行 PR 中变更过的测试文件。
- 推荐动作: 此 PR 设计清晰, 解决了实际痛点, 测试和文档完备。建议团队成员认真阅读 changed_test_files 和 _move_changes_dispatch 的实现, 理解其对测试选择的精确控制。对于未来 CI 工具的增强, 可考虑将此模式 (查询 PR 变更文件并过滤) 推广到其他命令。

功能与动机

现有 /rerun-test 命令虽然支持多路径, 但当 PR 涉及多个测试文件时, 作者仍需手动复制所有路径。本 PR 通过添加 --changed 标志, 实现一条命令即可重运行所有变更测试文件, 提升 CI 调试效率。该需求源于 PR #37058 等多文件修改场景。

实现拆解

1. 引入新标志与解析逻辑: 在 scripts/ci/utils/slash_command_handler.py 中定义 CHANGED_TESTS_FLAG (--changed) 和 CHANGED_TESTS_SHORT_FLAG (-c), 并在 main() 中从命令行参数中剥离它们, 将布尔值 include_changed_tests 传递给核心处理函数。
2. 实现变更测试文件查询: 新增 changed_test_files(pr) 函数, 通过 PR API 获取文件列表, 并使用 _is_rerunnable_test_path 过滤出符合条件的测试文件。核心筛选逻辑包括:
 - 文件状态非 removed;
 - 路径以 test_ 开头且在测试根目录下;
 - 对于多模态测试文件, 额外检查文件内是否定义了可收集的测试 (_collects_pytest_tests) 并排除 manual/ 目录。
3. 处理移动 (rename) 边界情况: 新增 _move_changes_dispatch 函数, 判断一个纯重命名是否改变了测试的调度路径 (例如从非 CI 根目录移入, 或跨越多模态测试边界)。纯移动且不影响调度的将被过滤掉。
4. 集成与错误处理: 在 handle_rerun_test 中, 当 include_changed_tests 为真时, 调用 changed_test_files 并将结果追加到 test_specs 列表。若最终列表为空且无其他明确规格, 则返回明确的错误提示。
5. 测试与文档配套: 在 test/registered/unit/tools/test_slash_command_handler.py 中新增 TestChangedTestFiles 测试类, 通过模拟 PR 对象覆盖多种文件状态组合 (修改、删除、移动、多模态等), 验证过滤逻辑。同时更新 docs/docs/developer_guide/contribution_g

uide.mdx 文档，说明新标志用法。

关键文件：

- scripts/ci/utils/slash_command_handler.py（模块 CI 脚本；类别 infra；类型 infrastructure；符号 `_is_rerunnable_test_path`, `_collects_pytest_tests`, `_move_changes_dispatch`, `changed_test_files`）：核心变更文件，实现了 `--changed` 标志的解析、变更测试文件查询和集成逻辑。
- test/registered/unit/tools/test_slash_command_handler.py（模块 测试脚本；类别 test；类型 test-coverage；符号 `TestChangedTestFiles`, `test_only_dispatchable_changed_test_files_are_selected`）：新增单元测试，覆盖了 `--changed` 标志的各种边界情况。
- docs/docs/developer_guide/contribution_guide.mdx（模块 贡献指南；类别 docs；类型 documentation）：更新文档，说明 `--changed` / `-c` 标志的用法。

关键符号：`changed_test_files`, `_is_rerunnable_test_path`, `_collects_pytest_tests`, `_move_changes_dispatch`

关键源码片段

scripts/ci/utils/slash_command_handler.py

核心变更文件，实现了 `--changed` 标志的解析、变更测试文件查询和集成逻辑。

```
# 定义新增标志常量
CHANGED_TESTS_FLAG = "--changed"
CHANGED_TESTS_SHORT_FLAG = "-c"

def _is_rerunnable_test_path(path):
    """判断一个仓库相对路径是否是一个可重运行的测试文件。"""
    # 检查是否在测试根目录下（test/registered/ 或多模态测试目录）
    under_test_root = path.startswith("test/registered/") or path.startswith(
        MULTIMODAL_TEST_DIR + "/"
    )
    # 基础条件：在测试根目录下、以 test_ 开头、以 .py 结尾
    if (
        not under_test_root
        or not os.path.basename(path).startswith("test_")
        or not path.endswith(".py")
    ):
        return False
    # 对于非多模态测试文件，满足基础条件即可
    if not path.startswith(MULTIMODAL_TEST_DIR + "/"):
        return True
    # 对于多模态测试文件，额外排除 manual/ 目录并检查文件内是否定义了测试
    return "manual" not in path.split("/") and _collects_pytest_tests(path)

def changed_test_files(pr):
    """返回 PR 中新增或修改的、可重运行的测试文件列表。"""
    return sorted(
```

```

f.filename
for f in pr.get_files()
if f.status != "removed" # 跳过已删除文件
and _is_rerunnable_test_path(f.filename) # 路径过滤
and (
    f.changes > 0 # 有内容变更
    or ( # 或者是影响调度的纯移动
        f.status == "renamed"
        and _move_changes_dispatch(f.previous_filename, f.filename)
    )
)
)
)

```

评论区精华

核心讨论集中在两个方面：

1. 移动 (rename) 文件的处理策略: Reviewer alisonshao 指出, 纯移动文件 (changes=0) 不应全部纳入重运行, 否则会像 commit ba214ef 那样导致 63 个文件被选中。作者回应称移动可能包含代码变更, 应运行测试。最终结论是: 仅当移动改变了文件的调度路径 (如进入 CI 根目录、跨越多模态边界) 时才纳入。
 2. 多模态辅助文件的过滤: alisonshao 指出 python/sglang/multimodal_gen/test/test_utils.py 等辅助文件会被错误选中, 导致 pytest 收集到零测试而退出码 5。解决方案是: 对多模态文件额外检查文件内是否定义了 def test_ 或 class Test, 并排除 manual/ 目录下的文件。
 3. Fork PR 的已知限制: Codex 机器人指出, 在 fork PR 中, 若作者无写权限, 工作流会 checkout main 而非 PR head, 导致新增测试文件不存在于本地, 解析时会报错。团队认为这与显式指定新增文件的行为一致, 暂不处理。
- 移动 (rename) 文件的处理策略 (design): 最终方案: 仅当移动改变了文件的调度路径 (如进入 CI 根目录、跨越多模态边界、在多模态子目录间移动) 时才纳入重运行。
 - 多模态辅助文件的过滤 (correctness): 对多模态文件额外检查文件内是否定义了 def test_ 或 class Test, 并排除 manual/ 目录下的文件。
 - Fork PR 中新增文件的处理限制 (question): 团队认为这与显式指定新增文件的行为一致, 属于已知限制, 暂不处理。

风险与影响

- 风险:
 1. 回归风险: 路径过滤逻辑 _is_rerunnable_test_path 被同时用于 glob 扩展和 --changed 查询, 修改可能影响 glob 的现有行为。但本次重构将过滤逻辑提取为公共函数, 且测试覆盖了多种边界情况, 风险较低。
 2. 功能边界模糊: --changed 仅覆盖测试文件本身被修改的情况, 不覆盖因源码修改而“受影响”的测试文件。这是设计范围内的明确限制, 但可能需要在文档中进一步强调。
 3. 性能影响: changed_test_files 需要遍历 PR 所有文件并读取多模态文件内容进行检查, 对于超大 PR (数千文件) 可能带来轻微延迟, 但通常 CI 脚本可接受。- 影响: 该 PR 显著提升了开发者的 CI 调试体验, 尤其对于涉及多个测试文件的 PR。影响范围限于

CI 工具链，不影响核心推理逻辑。团队成员在触发测试重运行时将减少手动操作，节省时间。同时，提取公共路径过滤函数增强了代码可维护性。 - 风险标记：CI 工具链变更，边界情况处理，Fork PR 限制

关联脉络

- PR #37058 未知：PR body 中提及此 PR 修改了三个测试文件，是此功能设计的直接动机来源。