

PR #24023 完整报告

sgl-project/sglang

ci: add /rerun-group to rerun all registered tests in a group

合并时间: 2026-04-30 01:24

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24023>

执行摘要

- 一句话: 添加 /rerun-group CI 命令, 按测试组批量重跑
- 推荐动作: 该 PR 作为一个 CI 工具改进, 设计上提取公共权限检查、复用 dispatch 是值得肯定的。但是非递归 glob 问题需要修复, 建议在合并前或后续 PR 中处理。整体上值得关注其设计模式, 尤其是权限检查提取和组名解析的通用化思路。

功能与动机

PR 描述指出: “Current CI only supports /rerun-state and /rerun-test. This PR adds support for validating all CI related to a specific feature.” 即开发者需要一种简单的方式验证某个特性 (如 HiCache、推测解码) 的全部测试, 而不是一个一个地手动指定测试文件。

实现拆解

1. 组名解析: 在 `slash_command_handler.py` 中新增 `_known_test_groups()` 和 `resolve_test_group_specs()` 函数。`_known_test_groups()` 扫描 `test/registered/` 下的所有子目录作为已知组名; `resolve_test_group_specs()` 根据用户输入的组名找到对应目录, 并 glob 匹配 `test_*.py` 文件列表 (当前使用非递归 pattern `'test_*.py'`)。
2. 权限检查提取: 将 `/rerun-test` 原有的 fork PR 权限检查逻辑抽取为独立的 `_check_rerun_test_permissions()` 函数, 使得 `/rerun-group` 可以复用, 避免了代码重复。
3. 调度函数: 新增 `handle_rerun_group()` 函数, 调用 `resolve_test_group_specs()` 获得文件列表, 然后通过现有的 batch dispatch 机制 (`_dispatch_batch`) 派发到相应的 runner 上。
4. workflow 触发: 在 `.github/workflows/slash-command-handler.yml` 的触发条件中添加了 `contains(github.event.comment.body, '/rerun-group')`, 使 GitHub Actions 能够识别该命令。
5. 文档更新: 在 `.claude/skills/ci-workflow-guide/SKILL.md` 中添加了 `/rerun-group` 命令的说明, 包括组名的定义和用法。

关键文件:

- `scripts/ci/utlils/slash_command_handler.py` (模块 CI 工具; 类别 infra; 类型 infrastructure; 符号 `_known_test_groups`, `resolve_test_group_specs`, `handle_rerun_test`, `_check_rerun_test_permissions`): 核心实现, 包含组名解析、调度、权限检查重构。

- `.github/workflows/slash-command-handler.yml` (模块 CI 工作流; 类别 `infra`; 类型 `infrastructure`) : 添加 `/rerun-group` 触发条件。
- `.claude/skills/ci-workflow-guide/SKILL.md` (模块 文档; 类别 `docs`; 类型 `documentation`) : 更新 CI 命令文档。

关键符号: `resolve_test_group_specs`, `handle_rerun_group`, `_check_rerun_test_permissions`, `_known_test_groups`

关键源码片段

`scripts/ci/utils/slash_command_handler.py`

核心实现, 包含组名解析、调度、权限检查重构。

```
def resolve_test_group_specs(group_name):
    """Resolve a test group name into test spec list."""
    group_name = group_name.strip().strip('/')
    # security checks
    if not group_name or group_name.startswith('.') or '/' in group_name or '..' in group_name.split('/'):
        return [], 'Invalid test group.'
    group_dir = os.path.join('test', 'registered', group_name)
    if not os.path.isdir(group_dir):
        return [], f'Unknown test group `{group_name}`. Known: {_known_test_groups()}'
    # glob test files (non-recursive)
    test_files = sorted(glob.glob(os.path.join(group_dir, 'test_*.py')))
    return [os.path.relpath(p, 'test') for p in test_files], None
```

评论区精华

主要讨论集中在递归 `glob` 的问题。`gemini-code-assist` 和 `chatgpt-codex-connector` 均指出 `resolve_test_group_specs` 中使用的是非递归 `glob`, 无法匹配子目录中的测试, 可能导致某些组 (如 `openai_server`) 的测试被遗漏。建议改用 `**/test_*.py` 加 `recursive=True`。目前该问题未被修复, 是一个已知的局限。此外, 作者通过测试验证了命令的正确性, 例如 `/rerun-group hicache` 成功触发了 `HiCache` 相关测试。

- 递归 `glob` 以包含子目录测试 (`correctness`): 建议未被采纳, 当前版本仍使用非递归 `glob`, 存在遗漏子目录测试的风险。

风险与影响

- 风险:

1. 测试遗漏风险: 非递归 `glob` 导致 `/rerun-group` 不会自动包含子目录下的测试。对于将测试文件组织在子目录中的组 (如 `openai_server`), 执行 `/rerun-group openai_server` 可能不会重跑任何测试, 因为顶层没有 `test_*.py` 文件。这违背了“重跑所有测试”的预期, 是目前最主要的未解决问题。
2. 组名敏感性: 组名必须与 `test/registered/` 下的目录名完全一致, 且区分大小写。用户如果使用不存在的组名, 会得到错误信息, 但不会触发重跑。这是设计如此, 不算严重风险。

- 影响：正面影响：为开发者提供快捷的批量重跑能力，便于验证特性测试集，提升 CI 使用效率。负面影响：如果组内测试组织在子目录中，当前命令可能只重跑顶层测试（如果顶层没有测试，会报无测试文件，不会静默失败，但可能少跑测试）。团队层面：鼓励测试文件按组件组织到 test/registered/<component>/ 目录下，并能够通过组名触发，有利于 CI 流程的标准化。

- 风险标记：非递归 glob 可能遗漏子目录测试，组名需与目录名完全一致

关联脉络

- 暂无明显关联 PR