

PR #24802 完整报告

sgl-project/sclang

slash command rerun UX: emoji semantics + result writeback

合并时间: 2026-05-09 18:19

原文链接: <http://prhub.com.cn/sgl-project/sclang/pull/24802>

执行摘要

- 一句话: 改进 CI 斜杠命令 emoji 语义并实现结果回写
- 推荐动作: 值得精读, 尤其是幂等设计和并发控制。建议后续类似的 CI 指令可用此模式。

功能与动机

根据 PR 描述, 原回复使用 🟢/🔴 容易被误解为测试通过 / 失败, 需要区分触发状态与测试结果; 此外, rerun 完成后用户无法及时获知结果, 需要回写机制。目的是改善开发者使用 slash-command 的体验。

实现拆解

1. 替换 emoji: 在 `slash_command_handler.py` 中将触发成功 / 失败的 emoji 替换为 🟢/🔴, 并保持错误提示用 🚫。
2. 新增结果回写脚本: `write_rerun_test_result.py` 是独立脚本, 通过 GitHub API 获取指定评论, 根据 marker (如 `<!--rrt:0-->`) 定位行, 替换 🟢 为 🟢/🔴, 并将 marker 改为: `done` 实现幂等。若 marker 未找到, 则按重试间隔 [0, 5, 15] 秒重试, 三次后静默退出, 不中断 CI。
3. 修改工作流: `rerun-test.yml` 新增 `reply_comment_id` 和 `reply_marker` 输入; 添加 `write-back-result` job, 在 `always()` 条件下根据测试 job 成功 / 失败决定状态, 调用写入脚本。job 通过 `concurrency: rerun-test-writeback-<id>` 串行化对同一评论的写回。
4. 修改 handler 触发流程: 在 `_dispatch_batch` 函数中添加 `reply_comment_id` 和 `reply_marker` 参数, 并传递到 workflow dispatch 的 inputs 中。handler 在触发前先创建一个占位评论, 获取 comment id, 并为每个 batch 分配唯一的 HTML 注释 marker (例如 `<!--rrt:0-->`), 然后将评论 body 编辑为包含这些 marker 的最终形式。
5. 幂等与容错: 写入脚本检测 marker 已存在则跳过; 重试策略避免短暂竞争; 最终失败不影响主流程 (返回 0)。

关键文件:

- `scripts/ci/utils/write_rerun_test_result.py` (模块 CI 脚本; 类别 infra; 类型 infrastructure; 符号 main): 核心新增文件, 实现评论内容更新逻辑, 包括 marker 查找、替换、重试和幂等退出。
- `scripts/ci/utils/slash_command_handler.py` (模块 CI 脚本; 类别 infra; 类型 infrastructure; 符号 `_dispatch_batch`): 修改了 emoji 映射和 `_dispatch_batch` 函数,

传递 reply_comment_id/reply_marker。

- .github/workflows/rerun-test.yml (模块 workflow; 类别 infra; 类型 infrastructure) : 新增 write-back-result job, 接收输入并调用脚本; 添加 reply_comment_id/reply_marker 输入。

关键符号: main, _dispatch_batch

关键源码片段

scripts/ci/utils/write_rerun_test_result.py

核心新增文件, 实现评论内容更新逻辑, 包括 marker 查找、替换、重试和幂等退出。

```
import argparse, os, sys, time, requests

RETRY_DELAYS_SEC = [0, 5, 15]

def main():
    ap = argparse.ArgumentParser()
    ap.add_argument("--comment-id", required=True, type=int)
    ap.add_argument("--marker", required=True, help="e.g. <!--rrt:0-->")
    ap.add_argument("--status", required=True, choices=["success", "failure"])
    ap.add_argument("--repo", required=True, help="owner/repo")
    args = ap.parse_args()

    token = os.environ.get("GITHUB_TOKEN")
    if not token:
        return 1

    icon = "👉" if args.status == "success" else "👎"
    done_marker = args.marker.replace("-->", ":done-->")
    headers = {
        "Authorization": f"Bearer {token}",
        "Accept": "application/vnd.github+json",
        "X-GitHub-API-Version": "2022-11-28"
    }
    url = f"https://api.github.com/repos/{args.repo}/issues/comments/{args.comment_id}"

    # 重试逻辑: 最多 3 次, 间隔逐步增加
    body = None
    for attempt, delay in enumerate(RETRY_DELAYS_SEC):
        if delay:
            time.sleep(delay)
        resp = requests.get(url, headers=headers, timeout=15)
        if resp.status_code != 200:
            return 1
        body = resp.json().get("body") or ""
        if done_marker in body:
            # marker 已被改写, 说明已成功回写, 幂等跳过
            return 0
```

```

if args.marker in body:
    break
print(f"Marker {args.marker} not found (attempt {attempt + 1}); will retry.")
else:
    # 三次重试后仍未找到 marker, 静默退出, 不将失败传播到 CI 流程
    print(f"WARNING: marker {args.marker} not found after retries; skipping writeback.")
    return 0

# 逐行替换: 找到含 marker 的行, 替换 🔄 为结果图标, 改写 marker 为 done_marker
new_lines = []
for line in body.splitlines(keepends=True):
    if args.marker in line:
        line = line.replace("🔄", icon, 1)
        line = line.replace(args.marker, done_marker)
    new_lines.append(line)
new_body = "".join(new_lines)

# 通过 PATCH 更新评论 body
update_resp = requests.patch(url, headers=headers, json={"body": new_body}, timeout=15)
if update_resp.status_code not in (200, 201):
    print(f"ERROR: PATCH failed: {update_resp.status_code} {update_resp.text}")
    return 1
return 0

```

评论区精华

无实质性讨论, 仅有一个 bot 确认变更的评论。主要设计决策在 PR body 中已有阐述。

- 暂无高价值评论线程

风险与影响

- 风险: 写回操作依赖 GitHub API, 若 token 不足或网络问题可能导致失败, 但脚本重试并返回 0 避免 CI 失败; 并发控制依赖于 job-level concurrency, 在极端情况下多个同时写回可能冲突, 但由于按行替换且 marker 不同, 冲突概率低; 另外如果 handler 在创建占位评论后失败, 占位评论残留, 但不会造成功能问题。
- 影响: 影响所有使用 /rerun-test 的 PR, 提升可见性和确定性; 对用户是正向体验提升; 对系统没有性能影响。
- 风险标记: 依赖 GitHub API 写入, 幂等设计降低失败影响, 低严重性

关联脉络

- 暂无明显关联 PR