

PR #25475 完整报告

sgl-project/sglang

pr-states: workflow_dispatch refresh on slash cmds

合并时间: 2026-05-16 18:34

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25475>

执行摘要

- 一句话: PR states workflow支持 workflow_dispatch 刷新
- 推荐动作: 该 PR 是纯基础设施改进, 逻辑清晰、改动适度, 值得合并。建议后续 PR 处理 review 中未解决的改进建议 (默认分支动态获取、覆盖 handle_rerun_test), 它们可进一步提升健壮性和覆盖率。

功能与动机

原 `pr-states.yml` 仅通过 `pull_request_target` 事件触发, 但 `slash` 命令触发的 workflow rerun 不会重新触发该事件, 导致 PR body 中的 CI 状态块无法自动刷新。PR body 指出: 'Make pr-states.yml refreshable via workflow_dispatch so /tag-and-rerun-ci and /rerun-failed-ci refresh the CI states block in PR body.'

实现拆解

1. 事件扩展: 在 `pr-states.yml` 的 `on` 配置中新增 `workflow_run` 事件, 监听 PR Test 和 PR Test Extra 工作流的 `requested` 和 `completed` 状态, 使 workflow rerun 也能触发状态刷新。
2. 并发控制优化: 将并发分组键从 `github.event.pull_request.number` 改为 `github.event.pull_request.head.sha || github.event.workflow_run.head_sha`, 避免同一 commit 的 `synchronize` 和 `workflow_run` 事件同时运行时产生 body 写入冲突。
3. PR 查找重构: 在 `job` 脚本中实现事件无关的 PR 编号查找逻辑——`pull_request_target` 直接从 `payload` 获取; `workflow_run` 优先使用 `pull_requests[]` 字段, 对 fork PR (该字段为空) 则通过 `head_repository.owner.login:head_branch` 格式反向查询 open PR, 并用 `head_sha` 匹配确保准确性, 若无匹配则 `cleanly skip`。
4. 状态块更新: 从 `payload` 中提前获取 PR 对象, 移除原脚本末尾的重复 `pulls.get` 调用, 利用 `pr.head.sha` 和 `pr.labels` 生成 CI 状态文本并写入 PR body。

关键文件:

- `.github/workflows/pr-states.yml` (模块 CI workflow; 类别 `infra`; 类型 `infrastructure`): 主要变更文件, 修改了 workflow 的触发事件、并发控制和 PR 查找逻辑, 支撑 `slash` 命令后的状态刷新。

关键符号: 未识别

关键源码片段

`.github/workflows/pr-states.yml`

主要变更文件，修改了工作流的触发事件、并发控制和 PR 查找逻辑，支撑 slash 命令后的状态刷新。

```
# .github/workflows/pr-states.yml ( 关键片段 )
name: PR States
on:
  pull_request_target:
    types: [opened, synchronize, reopened, labeled, unlabeled]
  workflow_run:
    # 监听 pr-test* 工作流的生命周期，使 workflow rerun 也能触发状态更新
    workflows: ["PR Test", "PR Test Extra"]
    types: [requested, completed]

concurrency:
  # 使用 head_sha 作为分组键，避免同一 commit 的 synchronize 和 workflow_run 事件同时写入
  # body
  group: pr-states-${{ github.event.pull_request.head.sha || github.event.workflow_run.head_sha
  }}
  cancel-in-progress: false

jobs:
  update:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/github-script@v7
        with:
          script: |
            // 事件无关的 PR 编号查找:
            // - pull_request_target: 直接从 payload 获取
            // - workflow_run: 优先使用 pull_requests[], 对 fork PR 则反向查询
            let prNumber;
            if (context.payload.pull_request) {
              prNumber = context.payload.pull_request.number;
            } else {
              const wr = context.payload.workflow_run;
              if (wr.pull_requests && wr.pull_requests.length > 0) {
                prNumber = wr.pull_requests[0].number;
              } else if (wr.head_repository && wr.head_branch) {
                const headSpec = `${wr.head_repository.owner.login}:${wr.head_branch}`;
                const { data: prs } = await github.rest.pulls.list({
                  owner: context.repo.owner,
                  repo: context.repo.repo,
                  state: 'open',
                  head: headSpec,
                });
                // 通过 head_sha 精确匹配，避免分支复用导致错误
```

```

const match = prs.find(p => p.head.sha === wr.head_sha) || prs[0];
if (!match) {
  core.info(`No open PR for head=${headSpec}; skipping.`);
  return;
}
prNumber = match.number;
} else {
  core.info(`workflow_run has no PR linkage; skipping.`);
  return;
}
}
// 获取最新 PR 数据, 更新 body 中的 CI 状态块
const { data: pr } = await github.rest.pulls.get({
  owner: context.repo.owner,
  repo: context.repo.repo,
  pull_number: prNumber,
});

```

评论区精华

1. 默认分支硬编码问题: gemini-code-assist[bot] 指出原实现中 `_dispatch_pr_states_refresh` 函数对 `dispatch ref` 硬编码了 `main` 字符串, 建议使用 `gh_repo.default_branch` 以支持不同默认分支的仓库。
 2. 覆盖不全问题: gemini-code-assist[bot] 建议在 `handle_rerun_test` 函数末尾也调用 `_dispatch_pr_states_refresh`, 因为 `/rerun-test` 和 `/rerun-group` 命令会触发新的 `pr-test*` 运行, 刷新 CI 状态块能确保 PR body 准确反映最新状态。这两个审查意见均未被合并, 后续可能有独立修复。
- 默认分支硬编码问题 (design): 未在本次 PR 中解决, 将在后续修复。
 - `handle_rerun_test` 未触发刷新 (design): 未在本次 PR 中解决, 将在后续修复。

风险与影响

- 风险:
 - 反向查询失败风险: fork PR 反向查询依赖于 `head_repository.owner.login` 和 `head_branch` 字段, 若 GitHub API 返回格式异常或分支被删除, 可能导致查找失败, 但已通过 `cleanly skip` 保证流程不中断。
 - 并发边界 case: 并发分组基于 `head_sha`, 若同一 SHA 对应多个 PR (如分支复用), 可能出现并发写入, 但概率极低。
 - 仅影响 CI 状态显示: 不涉及任何运行时逻辑, 无性能或安全性风险。
- 影响:
 - 用户 / 开发者: `slash` 命令后 PR body 中的 CI 状态块能自动更新, 减少人工检查工作。
 - 系统: 新增 `workflow_run` 事件监听和反向查询 API 调用, 会略微增加 GitHub Actions 的调用量, 但频率低、影响可忽略。
 - 团队: 该变更仅影响 `.github/workflows/pr-states.yml` 一个文件, 改动小、易于 review。

- 风险标记: 反向查询 fork PR 可能失败 (已处理), review 中两个建议未解决

关联脉络

- PR #25452 [CI] Show distinct run-name per event in pr-test workflow: 同为 CI workflow 改进, 调整了 pr-test workflow 的命名, 本 PR 通过 workflow_run 监听该 workflow。