

PR #25812 完整报告

sgl-project/sglang

pr-states: dispatch from pr-test* notify job (fix rerun status)

合并时间: 2026-05-20 06:57

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25812>

执行摘要

- 一句话: 修复 CI 重跑后 PR 状态不刷新
- 推荐动作: 对于维护 CI/CD 基础设施的工程师, 本 PR 有较高的参考价值——展示了如何利用 `workflow_dispatch` 绕过 `workflow_run` 在重跑时的限制, 以及 fork PR 兼容性考虑。可精读 `pr-states.yml` 中的 JavaScript 逻辑和 `notify job` 的设计。

功能与动机

PR body 指出 'Fix pr-states.yml not refreshing after a rerun (UI or slash command).

Root cause: GitHub does NOT re-fire `workflow_run.completed` for rerun attempts'. 因此需要让 `pr-states` 在重跑后也能更新状态。

实现拆解

1. 新增 `notify job`: 在 `pr-test.yml` 和 `pr-test-extra.yml` 末尾分别添加 `notify-pr-states job`。当 CI 运行完成 (无论成功失败) 且是自仓库的原生 PR 时, 通过 `gh workflow run` 命令向 `pr-states.yml` 发送 `workflow_dispatch` 事件, 传递 PR 编号。该 `job` 依赖聚合器 `job` (`pr-test-finish / pr-test-extra-finish`), 聚合器检查所有下游 `job` 是否有失败。
2. 调整 `pr-states` 触发器: 在 `pr-states.yml` 的 `on` 中添加 `workflow_dispatch` 输入参数 `pr_number`, 并保留 `pull_request_target` 和 `workflow_run` 作为备用 (尤其是 fork PR 场景)。调整并发组 `key` 以兼容三种事件源, 防止 `body PATCH` 竞态。
3. 更新渲染脚本: 在 `actions/github-script` 步骤中增加对 `workflow_dispatch` 事件的处理分支, 通过 `context.payload.inputs.pr_number` 获取 PR 编号。同时优化注释和逻辑, 明确各事件源的分支。

这些变更确保每次 CI 运行 (包括重跑) 完成后, `pr-states` 都能被触发去更新 PR body 中的状态区块。fork PR 不受 `notify job` 的影响 (token 只读), 依赖原有的 `workflow_run` 订阅实现初始更新。

关键文件:

- `.github/workflows/pr-states.yml` (模块 CI 工作流; 类别 `infra`; 类型 `infrastructure`): 触发事件重构, 新增 `workflow_dispatch` 支持, 更新 JavaScript 逻辑, 是修复的核心文件
- `.github/workflows/pr-test-extra.yml` (模块 CI 工作流; 类别 `infra`; 类型 `infrastructure`): 新增 `notify-pr-states job` 和聚合器, 展示工作流调度模式

- `.github/workflows/pr-test.yml` (模块 CI 工作流; 类别 `infra`; 类型 `infrastructure`) : 镜像 `pr-test-extra` 添加 `notify job`, 确保 `base` 测试也触发状态刷新

关键符号: 未识别

关键源码片段

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

触发事件重构, 新增 `workflow_dispatch` 支持, 更新 JavaScript 逻辑, 是修复的核心文件

```
# .github/workflows/pr-states.yml 事件触发器
on:
  pull_request_target:
    types: [opened, synchronize, reopened, labeled, unlabeled]
  workflow_run:
    workflows: ["PR Test Base", "PR Test Extra"]
    types: [requested, completed]
  workflow_dispatch: # 新增: 用于接收通知 job 的调度
  inputs:
    pr_number:
      description: 'PR number to refresh'
      required: true
      type: string

// 根据事件源获取 PR 编号
let prNumber;
if (context.payload.pull_request) {
  // pull_request_target: 直接从 payload 获取
  prNumber = context.payload.pull_request.number;
} else if (context.eventName === 'workflow_run') {
  // workflow_run: 尝试从 workflow_run 对象的 pull_requests 中获取
  const wr = context.payload.workflow_run;
  if (wr.pull_requests && wr.pull_requests.length > 0) {
    prNumber = wr.pull_requests[0].number;
  } else {
    // fork PR 的分支不在 default branch, 需要按 head 引用反向查找
    const [owner, branch] = wr.head_branch.split(':');
    // ... 反向查找逻辑 (省略)
  }
} else if (context.eventName === 'workflow_dispatch') {
  // workflow_dispatch: 由输入参数直接传入
  prNumber = context.payload.inputs.pr_number;
}

// 之后使用 prNumber 调用 GitHub API 更新 PR body 中的 CI 状态块
```

评论区精华

未产生 review 讨论, PR 合并者为作者本人, 表明改动已充分自验证。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险包括：（1）fork PR 兼容性：notify-pr-states job 会排除 fork PR，依赖 workflow_run 订阅可能在某些边界下延迟更新。PR 已保留 workflow_run 并作为备用，风险可控。（2）workflow_dispatch 速率限制：GitHub 对 gh workflow run 有速率限制，但 PR 中 notify job 只触发一次，影响小。（3）并发竞态：concurrency 组基于 PR 编号或 SHA 去重，可防止同时多个事件覆盖 body。整体风险较低。
- 影响：影响范围：所有使用 CI 状态显示功能的 PR（包括 fork PR）的刷新行为。对用户而言，重跑后 PR 状态块会正确更新，提升开发体验。对系统无明显性能影响。对团队：CI 工作流程架构增强了对 GitHub 事件限制的理解。
- 风险标记：CI 基础设施变更，workflow_dispatch 速率限制，fork PR 兼容性

关联脉络

- 暂无明显关联 PR