

PR #51095 完整报告

vllm-project/vllm

[CI] Fix CI authorization notification fallback

合并时间: 2026-08-05 16:34

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51095>

执行摘要

- 一句话: 修复 CI 授权通知竞态, 从 head commit 兜底解析 PR
- 推荐动作: 值得精读 `resolve_workflow_run_pr` 的设计: 它展示了 GitHub Actions `workflow_run` 事件中 PR 关联缺失时的标准兜底模式 (事件关联优先, `head_sha` 反查次之, 严格校验 `open` 状态和 head SHA)。关注点: 对象 `ready` 标签补发与 `marker` 去重如何配合, 以及 `workflow_run.id` 进入 `concurrency group` 对串行化的影响。该 PR 可作为 CI 事件竞态修复的参考样例。

功能与动机

PR body 明确指出根因: 在 #51093 上审批记录 workflow 虽然成功结束, 但其源运行携带的 `pull_requests` 数组为空, 因此后续通知作业被跳过; 随后添加 `ready` 标签时, 标签触发作业虽运行却因已存在可信审批而抑制评论, 错误假设审批路径已通知作者。作者还说明已从 #51093、`notify CI authorization workflow_run`、`ready label notification` 三个检索方向确认没有已存在的 PR 处理该问题。

实现拆解

1. 新增 `workflow_run` PR 解析兜底 (`.github/workflows/scripts/run_ci_command.py`): 新增 `GitHubClient.list_pulls_for_commit(commit)` 封装 `GET /repos/{repo}/commits/{sha}/pulls`, 对 `commit SHA` 做 URL 编码并对非列表响应抛 `ApiError`; 新增 `resolve_workflow_run_pr(workflow_run, github)` 及内部辅助 `find_matching_pr`, `main` 的 `workflow_run` 分支不再直接读取 `pull_requests[0]`, 而是先尝试事件自带关联, 再回退到按 `head_sha` 反查开放 PR, 且只接受 `state == "open"` 且 `head SHA` 精确匹配的 PR, 避免关联无关 PR。
2. 调整通知去重策略: `notify_authorized` 的 `labeled` 分支移除了 `has_trusted_approval` 短路条件, 使 `ready` 标签在审批通知漏发时可以补发通知; `CI_AUTHORIZED_COMMENT_MARKER` 标记去重仍保留, 正常场景不会重复通知。
3. 并发组与触发条件兜底 (`.github/workflows/notify-ci-authorized.yml`): `concurrency group` 在 `pull_request.number` 和 `workflow_run.pull_requests[0].number` 之后追加 `workflow_run.id` 作为最终兜底, 保证无 PR 号时每次运行拥有独立并发组; `job` 的 `if` 条件去掉 `pull_requests[0]` 依赖, 只要源 `workflow` 成功即触发, PR 解析交给脚本层。
4. 自托管 runner 迁移: `notify-ci-authorized.yml`、`add_label_automerge.yml`、`record-ci-approval.yml` 的 `runs-on` 从 `ubuntu-latest` 改为 `[self-hosted, linux, x64,`

vllm-runners], 与控制平面 workflow 迁移方向一致。

5. 测试配套 (.github/workflows/scripts/test_run_ci_command.py) : FakeGitHub 增加 pulls_for_commit 注入点和按 PR 号查找、缺失时抛 ApiError(404) 的 get_pr; 新增 test_workflow_run_resolves_pr_from_head_commit、test_workflow_run_ignores_unrelated_pr_association, 并将原 test_ready_label_does_not_notify_after_trusted_approval 反转为 test_ready_label_notifies_after_missed_approval_notification, 断言从“不通知”改为“通知一次且 @author”。

关键文件:

- .github/workflows/scripts/run_ci_command.py (模块 通知逻辑; 类别 infra; 类型 infrastructure; 符号 list_pulls_for_commit, resolve_workflow_run_pr, find_matching_pr) : 核心逻辑变更: 新增 resolve_workflow_run_pr 从 head commit 反查 PR, 调整 notify_authorized 的 labeled 分支以支持补发通知, main 的 workflow_run 分支改走新函数。
- .github/workflows/scripts/test_run_ci_command.py (模块 逻辑测试; 类别 test; 类型 test-coverage; 符号 list_pulls_for_commit, test_ready_label_does_not_notify_after_trusted_approval, test_ready_label_notifies_after_missed_approval_notification, test_workflow_run_resolves_pr_from_head_commit) : 测试配套: 扩展 FakeGitHub 支持按 commit 反查 PR, 新增两个 workflow_run PR 解析测试, 并反转 ready 标签相关测试的期望以覆盖补发场景。
- .github/workflows/notify-ci-authorized.yml (模块 通知 workflow; 类别 infra; 类型 configuration) : 配置侧兜底: concurrency group 增加 workflow_run.id fallback, job 触发条件不再要求 pull_requests[0], 并将 runs-on 迁移到 vllm-runners。
- .github/workflows/add_label_automerge.yml (模块 自动合并; 类别 infra; 类型 configuration) : 运行环境迁移至自托管 runner, 保持控制平面 workflow 一致。
- .github/workflows/record-ci-approval.yml (模块 审批记录; 类别 infra; 类型 configuration) : 运行环境迁移至自托管 runner, 配合整体 runner 策略。

关键符号: resolve_workflow_run_pr, find_matching_pr, list_pulls_for_commit, notify_authorized, main

关键源码片段

[.github/workflows/scripts/run_ci_command.py](#)

核心逻辑变更: 新增 resolve_workflow_run_pr 从 head commit 反查 PR, 调整 notify_authorized 的 labeled 分支以支持补发通知, main 的 workflow_run 分支改走新函数。

```
def resolve_workflow_run_pr(
    workflow_run: Mapping[str, Any],
    github: GitHubClient,
) -> dict[str, Any] | None:
    """从 workflow_run 事件解析关联的开放 PR。
```

当 `workflow\_run.pull\_requests` 为空或关联到无关 PR 时 (例如源 workflow

由非 PR 事件触发），这里会退化为按 head commit 反查 GitHub 上关联的开放式 PR。优先信任事件自带关联，其次才查 API，避免额外请求。

```
"""
```

```
head_sha = str(workflow_run.get("head_sha", ""))
```

```
if not head_sha:
```

```
    return None
```

```
# 记录已检查过的 PR 号，防止同一 PR 在两组候选里被重复处理
```

```
seen: set[int] = set()
```

```
def find_matching_pr(
```

```
    candidates: Sequence[Mapping[str, Any]],
```

```
) -> dict[str, Any] | None:
```

```
    for candidate in candidates:
```

```
        candidate_number = candidate.get("number")
```

```
        # 跳过非法或已检查过的 PR 号
```

```
        if not isinstance(candidate_number, int) or candidate_number in seen:
```

```
            continue
```

```
        seen.add(candidate_number)
```

```
        try:
```

```
            pr = github.get_pr(candidate_number)
```

```
        except ApiError as error:
```

```
            # PR 可能已被删除或不可见，跳过即可
```

```
            if error.status == 404:
```

```
                continue
```

```
            raise
```

```
            # 只接受仍处于 open 状态且 head SHA 完全匹配的 PR
```

```
            if pr["state"] == "open" and pr["head"]["sha"] == head_sha:
```

```
                return pr
```

```
    return None
```

```
# 第一优先: workflow_run 自带关联 (最可靠, 零额外 API 请求)
```

```
associated_pr = find_matching_pr(workflow_run.get("pull_requests") or [])
```

```
if associated_pr is not None:
```

```
    return associated_pr
```

```
# 第二优先: 按 head_sha 反查当前分支关联的 PR
```

```
return find_matching_pr(github.list_pulls_for_commit(head_sha))
```

[.github/workflows/notify-ci-authorized.yml](#)

配置侧兜底: concurrency group 增加 workflow_run.id fallback, job 触发条件不再要求 pull_requests[0], 并将 runs-on 迁移到 vllm-runners。

```
# concurrency group 使用三级 fallback:
```

```
# pull_request.number —— PR 事件
```

```
# workflow_run.pull_requests[0].number —— 有 PR 关联时
```

```
# workflow_run.id —— pull_requests 为空时的兜底, 保证组名唯一
```

```
concurrency:
```

```
  group: >-
```

```
    notify-ci-authorized-${{ github.event.pull_request.number
```

```
ll github.event.workflow_run.pull_requests[0].number
ll github.event.workflow_run.id }}
cancel-in-progress: false
```

评论区精华

本 PR 没有人工 review 讨论记录。唯一的 review 来自 claude[bot]：该 PR 来自 fork，自动 review 被禁用，维护者可评论 `@claude review` 触发一次性 review。实质的技术讨论主要体现在 PR body 的 Root cause 部分：作者指出 #51093 上 approvals 记录成功但其源运行 `pull_requests` 为空，随后 `ready` 标签触发作业却因 trusted approval 抑制评论，形成“双方都以为对方已通知”的竞态；并给出了 duplicate check 结果，确认没有既有 PR 解决该问题。

- 根因：workflow_run.pull_requests 为空导致通知跳过 (correctness)：引入 resolve_workflow_run_pr 按 head commit 反查开放 PR，并让 ready 标签在审批通知漏发时补发，同时保留 CI_AUTHORIZED_COMMENT_MARKER 去重。
- Fork PR 自动 review 被禁用 (other)：无技术结论，等待维护者手动处理。

风险与影响

- 风险：
 - GitHub API 时序依赖：list_pulls_for_commit 反查依赖 GitHub 对 commit-to-PR 关联的索引一致性，极端情况下（workflow 刚触发、commit 刚推送）仍可能查不到 PR，通知依旧跳过。但这是兜底路径，主路径 pull_requests 优先。
 - 通知去重语义变化：labeled 分支移除 trusted approval 短路后，若 CI_AUTHORIZED_COMMENT_MARKER 标记被删除或注释因故丢失，可能重复通知；正常情况 marker 检查仍在，风险可控。
 - 并发组行为变化：新增 workflow_run.id 后，极端情况下每个 notification run 有独立并发组，配合 cancel-in-progress: false 不会取消已运行作业，但同一 PR 的多次通知运行不再共享取消语义。
 - 自托管 runner 依赖：三个工作流迁移到 vllm-runners 池后，若池负载过高或网络异常，通知 / 标签 / 审批记录流程可能排队或失败；与现阶段的 runner 迁移方向一致。
 - 测试盲区：新增测试未覆盖 GitHub API 非 404 失败（如 5xx、超时）时 resolve_workflow_run_pr 的异常传播路径。
 - 影响：对贡献者：修复了 fork 场景下 approval 后收不到 CI 授权通知的问题，ready 标签可作为补发手段，提升 CI 授权链路可见性。对 CI 系统：notify-ci-authorized 不再依赖 workflow_run.pull_requests 数组，PR 解析稳定性提升；并发组兜底降低多触发间的互相干扰。对团队：CI 维护者需理解新增的 head commit 反查逻辑和三级 concurrency fallback；三个工作流迁移到自托管 runner，与 #51127 等 runner 迁移保持一致，运维边界更集中。
 - 风险标记：CI 通知核心链路变更，依赖 GitHub API 时序，通知去重逻辑调整，自托管 runner 依赖

关联脉络

- PR #51087 [CI] Add run-all comment commands: 同样修改 run_ci_command.py 及 CI 命令工作流, 属于同一 CI 命令处理链路的连续演进。
- PR #51079 [CI] Update CI notify workflow with PR write permissions: 直接修改同一个 notify-ci-authorized.yml 工作流的权限, 两处变更共同影响通知链路。
- PR #51127 [CI] Run control-plane workflows on vLLM runners: 同期的自托管 runner 迁移方向一致, 本 PR 也将三个授权工作流迁移到 vllm-runners。