

PR #38734 完整报告

sgl-project/sglang

[CI] Add /run-full-ci and /run-extra-ci slash commands

合并时间: 2026-09-10 04:56

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38734>

执行摘要

- 一句话: 为 CI 基础设施新增 /run-full-ci 和 /run-extra-ci 斜杠命令, 优化额外测试的触发流程。
- 推荐动作: 这是一个小而实用的 CI 流程改进。虽然代码变更不大, 但其设计决策 (复用权限、避免 fork 陷阱、rerun-only 策略) 体现了对 CI 系统细节的深入理解。建议 CI 维护者和频繁使用额外测试的贡献者精读相关实现和文档更新。

功能与动机

根据 PR body 描述, 当前额外 CI 只能通过 /tag-and-rerun-ci extra 触发, 无法单独运行。这意味着一个基线测试已通过的 PR, 为了运行额外测试, 不得不重新运行所有基线测试, 造成了不必要的资源浪费和时间延迟。新命令旨在提供更精细的控制。

实现拆解

1. 扩展斜杠命令处理器 (`scripts/ci/utils/slash_command_handler.py`): 新增 `EXTRA_WORKFLOW_NAME` 常量、`_run_extra_ci` 辅助函数和 `handle_run_extra_ci` 处理函数。`_run_extra_ci` 实现了仅重跑最新完成的 PR Test Extra 工作流的逻辑, 避免为不存在的运行或正在运行的流程创建重复任务。`handle_run_extra_ci` 则集成了标签添加和重跑逻辑, 并复用了 `can_tag_run_ci_label` 和 `can_rerun_failed_ci` 权限。主入口 `main` 函数扩展以识别 /run-full-ci、/run-extra-ci 及其 /rerun-* 别名。
2. 更新 GitHub Actions 工作流触发器 (`.github/workflows/slash-command-handler.yml`): 在 if 条件中添加新命令的字符串匹配, 确保新斜杠命令能触发该工作流。
3. 更新贡献者文档 (`docs/docs/developer_guide/contribution_guide.mdx`): 将新命令添加到 CI 命令列表, 并为其添加权限说明。同时修正了 /rerun-group 示例中的一个不存在的组名 (hicache -> rust-server)。
4. 更新权限说明文档 (`.github/FOLDER_README.md`, `.claude/skills/ci-workflow-guide/SKILL.md`): 同步更新权限密钥 `can_tag_run_ci_label` 和 `can_rerun_failed_ci` 所授权的命令列表, 以反映新命令。

关键文件:

- `scripts/ci/utils/slash_command_handler.py` (模块 CI 脚本; 类别 infra; 类型 infrastructure; 符号 `_run_extra_ci`, `handle_run_extra_ci`, `EXTRA_WORKFLOW_NAME`) : 核心逻辑变更文件, 实现了新斜杠命令的处理函数和主入口路由。

- docs/docs/developer_guide/contribution_guide.mdx (模块 贡献指南; 类别 docs; 类型 documentation) : 核心用户文档更新, 说明了新命令的用途、语法和权限。
- .github/workflows/slash-command-handler.yml (模块 workflows配置; 类别 config; 类型 configuration) : CI workflow触发器配置, 将新命令接入斜杠命令处理管道。
- .github/FOLDER_README.md (模块 配置文档; 类别 docs; 类型 documentation) : 权限配置说明文档, 同步更新了新命令所使用的权限密钥。
- .claude/skills/ci-workflow-guide/SKILL.md (模块 技能文档; 类别 docs; 类型 documentation) : AI 技能文档, 确保 AI 助手理解新的 CI 命令。

关键符号: `_run_extra_ci`, `handle_run_extra_ci`

关键源码片段

scripts/ci/utils/slash_command_handler.py

核心逻辑变更文件, 实现了新斜杠命令的处理函数和主入口路由。

```
# scripts/ci/utils/slash_command_handler.py
# 定义额外 workflow 名称常量, 与 GitHub Actions 中的 workflow name 字段一致。
EXTRA_WORKFLOW_NAME = "PR Test Extra"
```

```
def _run_extra_ci(gh_repo, pr):
```

```
    """
```

```
    为当前提交 SHA 运行 `PR Test Extra` 工作流。
```

```
    假设 `run-ci` 和 `run-ci-extra` 标签已打上。由于 `pr-test-extra.yml` 通过
    call-gate -> pr-gate.yml 动态获取标签, 因此可以安全地重跑一个之前因
    标签缺失而失败的运行。
```

```
    设计为重跑 (rerun-only), 不提供 workflow_dispatch 回退, 因为 extra 工作流
    的 dispatch 输入不包含 `pr_head_sha`, 对于 fork PR 会错误地测试 `main`。
    基本上每个 SHA 都有对应的 pull_request 运行, 所以只报告缺失情况, 而非
    强行创建。
```

```
    返回 (acted, message)。
```

```
    """
```

```
    # 获取当前 SHA 上所有名为 EXTRA_WORKFLOW_NAME 的工作流运行
```

```
    extra_runs = [
```

```
        run
```

```
        for run in gh_repo.get_workflow_runs(head_sha=pr.head.sha)
```

```
        if run.name == EXTRA_WORKFLOW_NAME
```

```
    ]
```

```
    if not extra_runs:
```

```
        # 如果不存在运行, 则提示用户推送提交以创建运行
```

```
        return (
```

```
            False,
```

```
            f"No `{EXTRA_WORKFLOW_NAME}` run exists for this commit, so there is "
```

```
            "nothing to re-run. Push a commit to create one.",
```

```
        )
```

```

# 采用与 handle_rerun_failed_ci 相同的最新运行优先规则，避免与正在运行的任务竞争 runner
run = _latest_run_per_workflow(extra_runs)[0]

if run.status != "completed":
    return (
        True,
        f"❓ [{EXTRA_WORKFLOW_NAME}]({run.html_url}) is already "
        f"{run.status} for this commit; left it alone.",
    )

try:
    if run.conclusion == "success":
        # 如果运行已成功，则执行完整重跑，因为 rerun_failed_jobs() 没有目标
        print(f" Full rerun of successful run {run.id}")
        run.rerun()
    else:
        try:
            # 否则，尝试仅重跑失败的作业
            print(f" rerun_failed_jobs on {run.id} ({run.conclusion})")
            run.rerun_failed_jobs()
        except Exception as e:
            # 如果 rerun_failed_jobs 失败，则回退到完整重跑
            print(f" rerun_failed_jobs rejected ({e}) - full rerun")
            run.rerun()
except Exception as e:
    return False, f"Failed to re-run [{EXTRA_WORKFLOW_NAME}]({run.html_url}): {e}"

return True, f"🔄 Re-running [{EXTRA_WORKFLOW_NAME}]({run.html_url})."

```

评论区精华

PR 的讨论区仅有机生文档预览评论和作者进行的命令测试，未发现技术争议或设计权衡讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更集中在 CI 基础设施脚本和文档，不影响推理核心逻辑。主要潜在风险是新命令的权限逻辑（复用现有权限）是否能被正确理解，但文档已同步更新。实现中明确避免了为 fork PR 触发错误的 dispatch，而是依赖已有的 pull_request 运行并进行重跑，这是一个稳健的设计。
- 影响：影响范围主要限于项目 CI/CD 流程。对贡献者而言，提供了更灵活、节省时间的 CI 触发方式，提升了开发体验。对系统而言，避免了不必要的基线测试重跑，可能节省 CI 资源。无用户可见的功能变更。
- 风险标记：CI 流程变更

关联脉络

- 暂无明显关联 PR