PR #23277 完整报告

sgl-project/sglang

[CI] Fix wait-for-jobs hanging when matrix job skipped at job level

合并时间：2026-04-21 02:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23277

执行摘要

修复了 GitHub Actions 中 wait-for-jobs 动作的一个关键缺陷：当矩阵作业（如 stage-a-test-cpu）因作业级 if: 条件被整体跳过时，等待动作会错误地持续轮询直至 240 分钟超时。通过精确检测 GitHub 生成的特定跳过条目模式并调整计数逻辑，使等待动作能快速成功，避免 CI 资源浪费和延迟。这是一个针对 CI 基础设施的低风险修复，不影响核心业务功能。

功能与动机

在 PR 仅修改内核文件等特定场景下，CI 中的矩阵作业（如 stage-a-test-cpu）会因作业级 if: 条件为假而被跳过。然而，现有的 wait-for-jobs 动作（用于 wait-for-stage-a 等）无法正确处理这种跳过，导致持续显示 found 1/4 jobs (waiting for more) 并挂起 4 小时。根本原因是 GitHub Actions 在作业级跳过时仅生成一个未展开的"跳过"条目（使用基础作业名前缀，无 (shard) 后缀），而非完整的矩阵条目，使得计数逻辑卡在 matchingJobs.length < spec.expected_count。此修复旨在消除这种不必要的等待，提升 CI 效率。

实现拆解

1. 变更入口：所有修改集中于 .github/actions/wait-for-jobs/action.yml，这是 GitHub Actions 自定义动作的定义文件，负责轮询并等待指定作业完成。
2. 核心逻辑改造：在检查 matchingJobs.length < spec.expected_count 的分支内，添加了对作业级跳过矩阵的检测逻辑。关键实现如下： - 检测条件：严格匹配单个条目、名称为基础前缀、状态为 completed、结论为 skipped，以避免误判部分展开的动态矩阵。 - 计数调整：将缺失的矩阵条目数（spec.expected_count - 1）加入总计数和完成计数，使等待条件得以满足。 - 日志输出：添加明确日志，便于调试和监控。
3. 测试与清理：修复方案通过本地测试工具验证了多种场景（如 bug 复现、全部成功、部分失败等），确保兼容性。在第二个提交中移除了测试工具，保持代码简洁。

.github/actions/wait-for-jobs/action.yml

这是唯一被修改的文件，包含了 wait-for-jobs 动作的核心逻辑，修复直接在此实现。

关键源码片段

.github/actions/wait-for-jobs/action.yml

这是唯一被修改的文件，包含了 wait-for-jobs 动作的核心逻辑，修复直接在此实现。

```
// 在检查 matchingJobs.length < spec.expected_count 的分支内
```

```
if (matchingJobs.length < spec.expected_count) {
    // 检测作业级跳过的矩阵：GitHub 会生成一个未展开的"跳过"条目
    // 条件：只有一个匹配作业，且其名称是基础前缀（无后缀），状态为完成，结论为跳过
    const unexpandedSkip = matchingJobs.length === 1 &&
                           matchingJobs[0].name === spec.prefix &&
                           matchingJobs[0].status === 'completed' &&
                           matchingJobs[0].conclusion === 'skipped';
    if (unexpandedSkip) {
        // 识别为作业级跳过：手动补全缺失的矩阵条目计数
        const missing = spec.expected_count - 1;
        totalCount += missing; // 增加总计数
        completedCount += missing; // 增加完成计数
        if (!cached) {
            console.log(`${spec.prefix}: job-level skip (bare entry, conclusion=skipped); treating as
            all ${spec.expected_count} skipped`);
        }
    } else {
        // 非跳过场景：继续等待更多作业
        console.log(`${spec.prefix}: found ${matchingJobs.length}/${spec.expected_count} jobs
        (waiting for more)`);
        allCompleted = false;
    }
}
```

评论区精华

无 review 评论或讨论。PR 由作者直接合并，表明变更较小且逻辑清晰，可能已通过内部验证。

风险与影响

- 技术风险：修复逻辑依赖于 GitHub Actions 生成跳过条目的特定行为，如果未来 GitHub 改变此行为（如生成多个跳过条目），可能导致检测失效。风险较低，因为 GitHub 的跳过模式相对稳定。严格的条件检查降低了误判风险。
- 影响范围：仅影响 CI 流水线中的等待动作，缩短矩阵作业被跳过时的 CI 执行时间，提升资源利用率。对 sglang 核心功能、性能或安全性无影响。

关联脉络

- 与 PR #23208（将 stage-a-test-cpu 拆分为 4 个矩阵分片）直接相关，两者都涉及同一 CI 作业的矩阵处理。
- 与近期多个 CI 修复 PR（如 #23201、#23053）同属基础设施优化范畴，反映了团队持续改进 CI 可靠性和效率的趋势。
- 此修复解决了矩阵作业跳过场景下的边缘 case，完善了 CI 等待机制，是 CI 流水线成熟度提升的一环。