PR #35392 完整报告

sgl-project/sglang

[CI] Skip fast-fail for scheduled stages

合并时间：2026-08-19 06:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35392

执行摘要

- 一句话：夜间测试加 scheduled 跳过快速失败，改为收集全部失败
- 推荐动作：值得 CI 维护者精读：这是一个用显式布尔输入替代 github.event_name 判定的典型范例，说明可复用工作流必须把“调用意图”作为显式契约传递，不能依赖事件类型猜测。变更单行、语义清晰、向后兼容，建议按当前方案合入；后续若出现更多跳过快速失败的场景，可考虑把条件集中封装到某个共享的 check-health 表达式或复合 action 中，避免条件继续膨胀。

功能与动机

PR body 明确指出根因：Release Branch Cut 通过 workflow_dispatch 调用 nightly 工作流，因此可复用阶段无法依赖 github.event_name == 'schedule' 判断是否为定时任务。虽然 nightly 工作流传入了 scheduled: true，但 SKIP_PR_TEST_HEALTH_CHECK 此前完全忽略该输入，导致 nightly 后续任务在早期失败后仍会级联 fast-fail，丢失后续失败信息。

实现拆解

1. 变更入口：.github/workflows/_pr-test-stage.yml 的 env 块，仅修改 SKIP_PR_TEST_HEALTH_CHECK 一行，其余配置不动。
2. 条件扩写：在原表达式 (skip_pr_test_health_check || test_parallel_dispatch || run_all_tests) 之前增加 inputs.scheduled，即“定时调度调用”也纳入跳过快速失败的范围。
3. 调用方语义：只有调用方工作流显式传入 scheduled: true 时才触发新行为；常规 PR 阶段不传该输入，保持原有 fast-fail 语义，因此向后兼容。
4. 配套验证：作者对常规 PR、scheduled、显式跳过、并行分发、run-all 五种输入组合做了人工评估，并运行 actionlint -ignore SC2086 .github/workflows/_pr-test-stage.yml 和仓库 pre-commit 检查；未新增自动化测试。

关键文件：

- .github/workflows/_pr-test-stage.yml（模块 CI 工作流；类别 infra；类型 infrastructure；符号 SKIP_PR_TEST_HEALTH_CHECK, inputs.scheduled）：唯一的变更文件，在 SKIP_PR_TEST_HEALTH_CHECK 条件中新增 inputs.scheduled，直接决定 nightly/weekly 可复用测试阶段是否会跨任务级联快速失败。

关键符号：SKIP_PR_TEST_HEALTH_CHECK

关键源码片段

.github/workflows/_pr-test-stage.yml

唯一的变更文件，在 SKIP_PR_TEST_HEALTH_CHECK 条件中新增 inputs.scheduled，直接决定 nightly/weekly 可复用测试阶段是否会跨任务级联快速失败。

```
env:
  # 冷缓存场景下默认超时较短，这里放宽到 300 秒
  HF_HUB_DOWNLOAD_TIMEOUT: 300
  HF_HUB_ETAG_TIMEOUT: 300
  # 跳过 PR 健康快速失败的核心条件（本次变更仅在这一行加入 inputs.scheduled）：
  # 1. inputs.scheduled：nightly/weekly 经 workflow_dispatch 调用时显式传入 true，
  # 因为此时 github.event_name 是 workflow_dispatch 而不是 schedule，不能靠事件名判断；
  # 2. caller_inputs 中的显式跳过 / 并行分发 / run-all 请求，保持原有跳过语义不变。
  SKIP_PR_TEST_HEALTH_CHECK: >-
    ${{ (inputs.scheduled
        || fromJson(inputs.caller_inputs).skip_pr_test_health_check
        || fromJson(inputs.caller_inputs).test_parallel_dispatch
        || fromJson(inputs.caller_inputs).run_all_tests) && 'true' || 'false' }}
  # 主分支（main）在维护模式下绕过测试的开关
  PR_TEST_BYPASS_MAINTENANCE_ON_MAIN: ${{ github.ref == 'refs/heads/main' && 'true' ||
  'false' }}
  USE_VENV: false
```

评论区精华

该 PR 无 review 评论和逐行讨论（comments_count=0，review_comments_count=0）。核心设计判断以 PR body 的 Root cause 与 Validation 形式记录：inputs.scheduled 是唯一可靠的定时任务信号，因为 github.event_name 在 workflow_dispatch 路径下不可用。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. CI 资源消耗上升：nightly/weekly 一旦有失败，将不再快速中止，而是跑完全部用例收集所有失败，CI 时长和资源占用会增加，这是有意的权衡。
  2. 调用方输入耦合：inputs.scheduled 依赖调用方显式传值；若某个调用方误传 scheduled: true，会意外关闭快速失败，但影响面仅限 CI 行为，不涉及推理运行时。
  3. 表达式兼容性：fromJson(inputs.caller_inputs) 与既有调用方式不变，inputs.scheduled 对未传布尔输入默认 false，不会破坏存量调用方。
  4. 验证盲区：无自动化测试覆盖该表达式，后续若再扩写条件，仍需靠人工评估和 actionlint 兜底。 - 影响：影响范围限于 CI/ 运维侧：夜间与周度测试在失败后能继续运行并汇总全部失败，显著提升 Release Branch Cut 等发布流程的失败可见性；常规 PR 测试的快速失败行为不变，开发者日常反馈节奏不受影响。对最终用户和推理服务无任何运行时影响。团队可借此减少“修完一个失败又冒出下一个失败”的循环，但需要接受 CI

资源的小幅增加。 - 风险标记：CI 行为变更，依赖调用方显式传入 scheduled，夜间失败将跑完全部用例，增加 CI 资源消耗，无自动化测试覆盖表达式

关联脉络

- PR #35238 Exclude multimodal-gen NPU jobs from fast-fail cascade: 同一 fast-fail 级联体系下的另一处收口：该 PR 在 NPU 单节点 /PR 测试阶段排除特定测试任务，本 PR 则在可复用阶段按 scheduled 输入整体关闭级联，二者共同演进 CI 失败聚合策略。
- PR #33685 [NPU CI] Reorganize test output/log directory structure with workflow context: 改动了同一家族的可复用测试阶段工作流（_npu-single-node-test-stage.yml 等），说明近期 CI 正在围绕 fast-fail、日志目录和套件模式做系统性整理，本 PR 是这条演进线上的一环。