PR #25266 完整报告

sgl-project/sglang

[AMD][CI] Clean up AMD nightly + pr-test workflows

合并时间：2026-05-21 14:30

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25266

执行摘要

- 一句话：AMD CI 工作流清理与修复
- 推荐动作：对于 CI 基础设施维护人员值得精读，展示了如何系统地清理和修复 CI 工作流：识别功能性 bug、统一命名、对齐逻辑、补充输入参数。特别是 run_all_tests 和 continue_on_error 的串联设计值得借鉴。对于一般开发者了解 AMD CI 结构也有帮助。

功能与动机

来自 PR body: '4 functional fixes + 5 consistency / refactor passes, in 9 small commits. Workflow and CI plumbing only — no model, kernel, or test-threshold change.' 主要动因包括：手动 dispatch 因缺少 run_all_tests 导致全部跳过；存在空 stage 浪费 CI 资源；35x 命名不一致；rocm720 的 continue_on_error 未对齐；以及 ensure_vram_clear.sh 路径错误导致 stage-c 始终失败。同时提高 GLM 评估样本数以获取更稳定的准确率信号。

实现拆解

关键源码片段

.github/workflows/pr-test-amd-rocm720.yml

包含 run_all_tests 输入、set-continue-on-error 步骤和 disaggregation 重命名等关键修复。

```
# pr-test-amd-rocm720.yml 新增 run_all_tests 输入，使 workflow_dispatch 可跳过路径过滤
on:
  workflow_dispatch:
    inputs:
      run_all_tests:
        description: '运行所有测试（跳过变更检测）。当设置 target_stage/target_stage_select
        时忽略。'
        required: false
        type: boolean
        default: false

# check-changes job 中新增 set-continue-on-error 步骤，
# 统一三种触发条件，输出 continue_on_error 供下游 job 使用
jobs:
  check-changes:
```

```
outputs:
  continue_on_error: ${{ steps.set-continue-on-error.outputs.continue_on_error }}
steps:
  - name: Set continue-on-error for schedule/full runs
    id: set-continue-on-error
    run: |
      # 当 run_all_tests 为 true、或 inputs.continue_on_error 为 true、
      # 或事件为 schedule 时启用 continue-on-error
      if [[ "${{ steps.run-mode.outputs.run_all_tests }}" == "true" \
            || "${{ inputs.continue_on_error }}" == "true" \
            || "${{ github.event_name }}" == "schedule" ]]; then
        echo "continue_on_error=true" >> $GITHUB_OUTPUT
        echo "Continue-on-error: ENABLED (run_all_tests=${{ steps.run-mode.outputs.run_all_
        tests }}, input=${{ inputs.continue_on_error }}, event=${{ github.event_name }})"
      else
        echo "continue_on_error=false" >> $GITHUB_OUTPUT
        echo "Continue-on-error: DISABLED"
      fi
```

实现步骤

1. 修复 workflow_dispatch 跳过问题：在 pr-test-amd.yml 和 pr-test-amd-rocm720.yml 的 workflow_dispatch 部分新增 run_all_tests 布尔输入，并在 check-changes job 中增加 || inputs.run_all_tests 回退逻辑，使得手动触发时可以绕过路径过滤直接运行所有测试。
2. 移除空 stage：从 pr-test-amd.yml、pr-test-amd-rocm720.yml、test/run_suite.py 的 PER_COMMIT_SUITES 以及 .claude/skills/write-sglang-test/SKILL.md 中删除 stage-b-test-1-gpu-small-amd-mi35x（该套件注册的测试文件位于 test/manual/，run_suite.py 不会扫描该目录）。
3. 统一命名规范：将 stage-b-test-large-8-gpu-35x-disaggregation-amd 及其相关引用（工作流文件、测试文件中的 register_amd_ci 调用、run_suite.py）中的 35x 替换为 mi35x，保持与现有 mi35x 前缀一致。
4. 对齐 continue_on_error 逻辑：在 pr-test-amd-rocm720.yml 的 check-changes job 中添加 set-continue-on-error 步骤，统一处理 schedule/run_all_tests/ 显式输入三种场景，并将所有测试 job 的 condition 从内联检查改为引用 needs.check-changes.outputs.continue_on_error。
5. 修复路径错误：将 stage-c-test-4-gpu-amd-rocm720 中错误的 scripts/ensure_vram_clear.sh 路径修正为 scripts/ci/amd/ensure_vram_clear.sh。
6. 提升 GLM 评估样本数：在 test_glm51_eval_amd.py、test_glm51_eval_mi35x.py、test_glm5_mxfp4_eval_mi35x.py 中将默认 GSM8K_NUM_QUESTIONS 从 200 改为 1319，使用完整的 GSM8K 测试集以获得更稳定的准确率指标。
7. 重新排序 nightly 工作流：nightly-test-amd.yml 和 nightly-test-amd-rocm720.yml 按模型分组排列 job 顺序，使 MI30x 与 MI35x 变体相邻，并添加章节注释，YAML 内容与 main 字节等效。

8. 冲突合并：两次合并 main 分支，处理了 pr-test-amd-rocm720.yml 中 timeout 变更和 nightly-* 中 NSA→DSA 重命名导致的冲突。

关键文件：

- .github/workflows/nightly-test-amd-rocm720.yml（模块 CI 配置；类别 infra；类型 infrastructure）：最大修改文件，重新排序 job 并添加分组注释，涉及 +524/-490 行变更。
- .github/workflows/nightly-test-amd.yml（模块 CI 配置；类别 infra；类型 infrastructure）：同样重新排序 job，+454/-409 行，与 rocm720 变体对齐。
- .github/workflows/pr-test-amd-rocm720.yml（模块 CI 配置；类别 infra；类型 infrastructure）：包含 run_all_tests 输入、set-continue-on-error 步骤和 disaggregation 重命名等关键修复。
- .github/workflows/pr-test-amd.yml（模块 CI 配置；类别 infra；类型 infrastructure）：添加 run_all_tests 输入并统一 disaggregation 阶段命名。
- test/run_suite.py（模块 测试调度；类别 test；类型 test-coverage）：更新 PER_COMMIT_SUITES 移除旧 stage 并添加 mi35x 命名。
- test/registered/amd/accuracy/mi30x/test_glm51_eval_amd.py（模块 模型评估；类别 test；类型 test-coverage）：提升 GSM8K 默认样本数至 1319 以提高准确率稳定性。

关键符号：未识别

评论区精华

唯一一次 review 评论来自 gemini-code-assist[bot]，指出 scripts/ci/utils/slash_command_handler.py 中 amd_stages 列表缺少 stage-b-test-large-8-gpu-35x-disaggregation-amd 和 stage-c-test-4-gpu-amd，并建议优化工作流自动定位逻辑以支持多个 AMD CI 工作流。作者未回应或采纳，PR 合并后该问题保留在代码中。HaiShaw 最终批准并合并 PR。

- AMD CI slash command handler 缺少 stage (design): 作者未在 PR 中回应或采纳，PR 合并后该问题仍然存在。

风险与影响

- 风险：尽管变更仅涉及 CI 基础设施，但存在以下风险： 1) 工作流定义错误可能导致 CI 提早退出或跳过必要测试，但每次 PR 的 CI 运行都能暴露问题； 2) 重新排序 nightly 工作流可能影响依赖这些 job 名称的外部监控或通知； 3) 默认样本数提升至 1319 会延长 GLM 评估运行时间，增加计算成本； 4) 两次 main 合并可能引入隐藏冲突，但已通过解决记录确认。总体风险较低。
  - 影响：直接受影响的是 AMD CI 维护团队和日常使用 AMD 工作流的所有开发者。修复了手动触发时工作流无法运行的问题，消除了一个空 stage 的资源浪费，提高了命名一致性，使 continue_on_error 行为在 rocm720 与标准 pr-test-amd 之间一致。GLM 评估获得更可靠准确率信号。不影响任何模型推理、训练或用户 API。
  - 风险标记：CI 工作流变更，工作流入口调整，测试样本数增加，合并冲突

关联脉络

- 暂无明显关联 PR