PR #22974 完整报告

sgl-project/sglang

[AMD] fix AMD CI gate

合并时间：2026-04-17 18:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22974

执行摘要

- 一句话：修复 AMD CI 门控逻辑，确保调度 / 全量测试模式下所有子测试都能执行。
- 推荐动作：此 PR 主要面向 CI 维护者和 AMD 平台开发者。建议关注 .github/workflows/pr-test-amd.yml 中 sgl-kernel-unit-test-amd 作业的条件逻辑和 run 步骤的变更，理解其如何通过 CONTINUE_ON_ERROR 环境变量区分 PR 模式与调度 / 全量模式。对于一般开发者，无需深入研读。

功能与动机

PR 标题和提交历史表明，此变更旨在修复 AMD CI 门控（gate）问题。具体来说，是为了解决在调度（schedule）或全量（full）运行模式下，CI 作业可能因部分测试失败而提前终止，导致后续测试无法执行的问题。这确保了 CI 在非 PR 模式下能更全面地收集测试结果。

实现拆解

1. 添加门控依赖：在 .github/workflows/pr-test-amd.yml 中，为 sgl-kernel-unit-test-amd 作业添加了对 call-gate 作业的依赖（needs: [check-changes, call-gate]），并修改其条件判断逻辑，确保只有当 call-gate 成功或跳过时才执行。
2. 引入聚合测试执行逻辑：在同一作业的 run 步骤中，新增环境变量 CONTINUE_ON_ERROR 和辅助函数 run_pytest。当 CONTINUE_ON_ERROR 为 true（即调度 / 全量模式）时，即使单个 pytest 命令失败，也会继续执行后续测试并累加失败计数；在 PR 模式下则保持快速失败行为。
3. 修正作业名称：将 stage-b-test-large-8-gpu-disaggregation-amd 重命名为 stage-b-test-large-8-gpu-35x-disaggregation-amd，使其与现有阶段命名模式（如 stage-c-test-large-8-gpu-amd-mi35x）保持一致。
4. 无测试或文档配套改动：此 PR 仅修改 CI 配置文件，未涉及源码、测试或文档的变更。

关键文件：

- .github/workflows/pr-test-amd.yml（模块 CI 流水线；类别 infra；类型 infrastructure）：这是唯一被修改的文件，包含了 AMD CI 流水线的核心配置，所有变更均在此文件中进行。

关键符号：未识别

关键源码片段

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

这是唯一被修改的文件，包含了 AMD CI 流水线的核心配置，所有变更均在此文件中进行。

```
# 片段来自 sgl-kernel-unit-test-amd 作业的配置
sgl-kernel-unit-test-amd:
  # 新增对 call-gate 作业的依赖，确保门控检查通过
  needs: [check-changes, call-gate]
  if: |
    always() && (
      (contains(format(',{0},', inputs.target_stage || inputs.target_stage_select), ',sgl-kernel-unit-test-amd,')) ||
      (
        !(inputs.target_stage || inputs.target_stage_select) &&
        # 新增条件：只有当 call-gate 成功或跳过时才执行
        (needs.call-gate.result == 'success' || needs.call-gate.result == 'skipped') &&
        needs.check-changes.outputs.sgl_kernel == 'true'
      )
    )
  steps:
    - name: Run test
      timeout-minutes: 14
      env:
        # 新增环境变量，用于区分运行模式
        CONTINUE_ON_ERROR: ${{ needs.check-changes.outputs.continue_on_error }}
      run: |
        # 在 continue-on-error 模式（调度/全量运行）下，持续运行所有 pytest 文件并聚合退出码。
        # 在 PR 模式下，保持快速失败行为。
        failures=0
        run_pytest() {
          if [[ "$CONTINUE_ON_ERROR" == "true" ]]; then
            "$@" || failures=$((failures + 1))
          else
            "$@"
          fi
        }
        # 使用辅助函数执行各个测试模块
        run_pytest docker exec -w /sglang-checkout/sgl-kernel/tests ci_sglang python3 -m pytest test_moe_align.py
        run_pytest docker exec -w /sglang-checkout/sgl-kernel/tests ci_sglang python3 -m pytest test_moe_topk_softmax.py
        # ... 其他测试命令
        # 最终根据聚合的失败计数退出
        if [[ $failures -gt 0 ]]; then
          exit 1
        fi
```

评论区精华

Review 评论中仅包含一条由 amd-bot 生成的自动化代码审查摘要，其中指出：

> 这些剩余的 !failure() && !cancelled() 位于依赖 wait-for-stage-a-amd 或 wait-for-stage-b-amd 的作业上，而这些等待作业本身已包含显式的 call-gate 检查。由于 always() 位于 if: 子句顶部，这些作业使用 (!failure() && !cancelled()) 来检查其上游等待作业是否失败——门控已通过传递性强制执行。这是正确的。

该评论确认了现有门控逻辑的正确性，但未对本次变更提出异议或深入讨论。PR 作者 yctseng0211 和合并者 bingxche 之间无实质性技术讨论。

- 门控逻辑正确性验证 (correctness): 审查确认现有逻辑正确，未对本次变更提出异议。

风险与影响

- 风险：1. CI 行为变更风险：在调度 / 全量模式下引入“继续执行”逻辑，可能导致 CI 作业在测试失败时仍显示为成功（如果仅看作业状态），需依赖聚合的退出码或日志分析来识别失败。 2. 依赖关系风险：新增对 call-gate 作业的依赖，若 call-gate 作业本身出现故障或配置错误，可能影响 sgl-kernel-unit-test-amd 作业的执行。 3. 命名一致性风险：重命名解聚测试作业可能影响现有 CI 触发脚本或监控工具，需确保相关引用已同步更新。 4. 低安全与兼容性风险：变更仅限 CI 基础设施，不涉及生产代码、数据或接口。
- 影响：1. 对团队的影响：CI 维护者需注意新的测试执行模式，在分析调度运行结果时需检查聚合失败计数而非仅依赖作业状态。 2. 对系统的影响：提升 CI 在非 PR 模式下的测试覆盖率，能更早发现潜在问题，但可能增加单次运行时长（因失败后仍继续）。 3. 对用户的影响：无直接影响，此为内部基础设施改进。 4. 影响范围：仅限于 AMD 平台的 CI 流水线，不影响其他平台（如 NVIDIA、CPU）或生产环境。
- 风险标记：CI 行为变更 , 依赖关系调整

关联脉络

- PR #23017 ci: install rust toolchain in ci_install_dependency.sh: 同属 CI 基础设施改进，涉及 CI 脚本和依赖安装。
- PR #22274 [AMD] CI Job Monitor: fix queue time, utilization, and summary metrics: 同属 AMD CI 相关修复，涉及 CI 监控脚本。
- PR #23040 [NPU]chore(docker): use editable install for sglang in npu.Dockerfile: 同属平台特定（NPU vs AMD）的基础设施配置调整。