PR #2231 完整报告

radixark/miles

fix(ci): cancel PR tests after closure

合并时间：2026-08-07 04:00

原文链接：http://prhub.com.cn/radixark/miles/pull/2231

执行摘要

- 一句话：PR 关闭后自动取消排队 / 运行中测试，释放 GPU runner
- 推荐动作：值得快速阅读（重点看 .github/workflows/pr-test.yml 的 diff 与新增测试）。它展示了一个值得借鉴的 CI 资源治理模式：用 closed 事件 + 同一 concurrency group 实现 cancellation-only run，并通过文本断言锁定 workflow 关键约定。建议后续把相同模式推广到 pr-test-rocm.yml。整体属于小而稳的修复，不需要深入评审。

功能与动机

PR body 明确描述了症状：合并或关闭的 PR 会留下排队或运行中的 PR Test 作业持续占用 GPU runner。根因有两点：pr-test.yml 没有订阅 pull_request.closed；也没有一个同 concurrency group 的 close run 来激活 cancel-in-progress。在 GPU runner 昂贵且紧张的背景下，这是一个直接造成资源浪费的 CI 缺陷。

实现拆解

变更入口是 .github/workflows/pr-test.yml，核心思路是让 PR 关闭事件触发一个“只取消、不干活”的 run。步骤拆解如下：

1. 订阅 closed 事件：pull_request.types 从 [opened, synchronize, reopened, ready_for_review, labeled] 扩展为包含 closed，PR 被合并或关闭时也触发一次 workflow run。
2. 统一并发键：concurrency group 的 key 从 github.event.pull_request.number 改为 github.event.number，保证 closed 事件与同一 PR 的普通事件落入同一组，cancel-in-progress: true 才能取消旧测试；schedule / dispatch 场景仍通过 || fallback 保持互不取消。
3. 关闭 run 跳过根作业：resolve-ci-policy、docker-paths 增加 if: github.event.action != 'closed'；docker-build 从 always() && !cancelled() 扩展为 always() && !cancelled() && github.event.action != 'closed'，避免关闭时仍解析策略、计算 diff 或构建镜像。
4. 测试锁定：tests/ci/test/test_run_suite.py 新增 test_closed_pr_only_cancels_existing_run，用字符串断言锁住 closed 订阅、concurrency key 格式、三个作业的 closed 守卫；原有 72 个测试全部通过。
5. 文档同步：docs/ci/00-stage.md 补充说明 closed 事件为 cancellation-only，共享并发键但不启动任何 resolver 或测试作业。

配套分析：本次没有修改训练 / 推理源码，属于纯 CI 基础设施调整；check-yaml hook 通过。

关键文件：

- .github/workflows/pr-test.yml（模块 工作流配置；类别 infra；类型 infrastructure）：核心变更文件：订阅 pull_request.closed 事件、统一 concurrency group key 为 github.event.number，并为 resolve-ci-policy、docker-paths、docker-build 三个作业增加 closed 事件守卫，使关闭触发的 run 只取消不执行。
- tests/ci/test/test_run_suite.py（模块 CI 测试；类别 test；类型 test-coverage；符号 test_closed_pr_only_cancels_existing_run）：新增 test_closed_pr_only_cancels_existing_run，用字符串断言锁定 closed 事件订阅、concurrency key 格式和三个关键作业的 closed 守卫，防止未来回归。
- docs/ci/00-stage.md（模块 CI 文档；类别 docs；类型 documentation）：同步文档，说明 pr-test.yml 将 pull_request.closed 视为 cancellation-only 事件，帮助后续维护者理解该设计。

关键符号：test_closed_pr_only_cancels_existing_run

关键源码片段

.github/workflows/pr-test.yml

核心变更文件：订阅 pull_request.closed 事件、统一 concurrency group key 为 github.event.number，并为 resolve-ci-policy、docker-paths、docker-build 三个作业增加 closed 事件守卫，使关闭触发的 run 只取消不执行。

```
# .github/workflows/pr-test.yml 关键片段
# 订阅 closed 事件：PR 关闭时触发一个专门用于取消的 run
on:
  pull_request:
    types: [opened, synchronize, reopened, ready_for_review, labeled, closed]

concurrency:
  # 用 github.event.number 统一 key：closed run 与同 PR 的普通 run 落入同一 group，
  # cancel-in-progress 才能取消排队 / 运行中的测试；schedule 时回退到 schedule / run_id
  group: ${{ github.workflow }}-${{ github.event.number || github.event.schedule || github.run_id
  }}
  cancel-in-progress: true

jobs:
  resolve-ci-policy:
    # closed run 跳过根作业：这次触发只负责 cancel，不执行任何测试逻辑
    if: github.event.action != 'closed'
    runs-on: ubuntu-latest

  docker-build:
    needs: [docker-paths]
    # always() 分支同样排除 closed，避免关闭 PR 时仍构建 GPU 镜像
    if: always() && !cancelled() && github.event.action != 'closed'
```

tests/ci/test/test_run_suite.py

新增 test_closed_pr_only_cancels_existing_run，用字符串断言锁定 closed 事件订阅、concurrency key 格式和三个关键作业的 closed 守卫，防止未来回归。

```
# tests/ci/test/test_run_suite.py 中新增的测试方法
def test_closed_pr_only_cancels_existing_run(self):
    workflow = self._workflow()
    # 锁定 closed 事件订阅：这是触发取消 run 的入口
    assert "types: [opened, synchronize, reopened, ready_for_review, labeled, closed]" in
    workflow
    # 锁定 concurrency key：closed run 必须与同 PR 的普通 run 同组，
    # cancel-in-progress 才会取消排队 / 运行中的测试
    assert (
        "group: ${{ github.workflow }}-${{ github.event.number || github.event.schedule || github.
        run_id }}"
        in workflow
    )
    # 锁定根作业守卫：三个关键作业在 closed 事件下都必须跳过
    for job_name in ("resolve-ci-policy", "docker-paths", "docker-build"):
        job_header = workflow.split(f"  {job_name}:", 1)[1].split("    runs-on:", 1)[0]
        assert "github.event.action != 'closed'" in job_header
```

评论区精华

该 PR 没有产生实质 review 评论，两位 reviewer（yushengsu-thu、austin362667）均直接 APPROVED。作者在 body 中主动给出 Review Focus：核查 closed 触发器与 github.event.number 并发键，以及每个根作业或 always() 作业是否跳过 closed 事件。仓库的 CI 工作流测试风格是用文本断言锁住 workflow 语义，本次新增的 test_closed_pr_only_cancels_existing_run 正是把这些约定固化为回归保护，弥补了没有人工讨论的不足。

- closed 事件 run 是否只取消不干活 (design): 通过 test_closed_pr_only_cancels_existing_run 断言 closed 订阅、统一并发键和 resolve-ci-policy / docker-paths / docker-build 三个作业的 closed 守卫，行为被锁死；docker-build 的 always() 分支也补上了 closed 排除。

风险与影响

- 风险：
  1. 并发键误伤风险：group 依赖 github.event.number；schedule / dispatch 事件该值为空，回退到 schedule 或 run_id，与改动前等价，风险低；但未来新增事件类型时需注意 fallback，避免意外共享 group 导致误取消。
  2. closed run 自身开销：每次关闭 PR 会新增一次 workflow run，虽然不启动作业，仍占用少量 Actions 配额和 runner 槽位；若未来新增根作业或 always() 作业时忘记加 closed 守卫，关闭时反而会启动新作业。
  3. 测试断言脆弱：test_run_suite.py 用字符串匹配锁定 workflow 文本，YAML 格式细节变化会导致误报或漏检，属于该测试模式的固有弱点。

4. 覆盖缺口：pr-test-rocm.yml（MI300X）未做同样处理，合并后的 ROCm PR 测试仍可能占用 AMD runner。

- 影响：
  - 成本与资源：直接消除合并 / 关闭 PR 后排队与运行中的 PR Test 对 NVIDIA GPU runner 的空耗，CI 队列更早排空。
  - 开发者体验：合入 PR 后无需再等待无关测试结束，队列释放更快。
  - 行为不变性：正常运行、schedule、manual dispatch 的测试范围与取消语义均不受影响，github.event.number 在 pull_request 事件中与原 key 等价。
  - 维护成本：新增一个行为契约测试，未来改动 workflow 时破坏该语义会被 CI 捕获；需要同步维护文档。
  - 风险标记：并发键 fallback 风险，closed run 排队开销，文本断言脆弱，ROCM 流程未覆盖

关联脉络

- PR #2230 fix(ci): disable MI300X runner jobs: 同为 CI 资源治理：该 PR 停用 MI300X 调度避免浪费 AMD runner，本 PR 在合并后取消排队测试避免浪费 NVIDIA runner，目标一致。
- PR #2217 test(ci): right-size session model GPU coverage: 调整 PR 测试 GPU 规格以匹配模型需求，与本次改动共同优化 PR Test 的 GPU 占用。
- PR #1606 ci(rocm): add ROCm CI workflow for MI300X self-hosted runners: 引入 pr-test-rocm.yml，与 pr-test.yml 同属 PR 测试工作流家族；本次模式未来可推广到 ROCm 工作流。