PR #2529 完整报告

radixark/miles

refactor(ci): gate PR image builds on CPU tests

合并时间：2026-08-17 13:36

原文链接：http://prhub.com.cn/radixark/miles/pull/2529

执行摘要

- 一句话：CPU 测试优先放行，PR 镜像构建后置门控
- 推荐动作：值得精读，适合负责 CI 流水线维护的工程师。重点关注三个设计点：可复用工作流提取如何降低主工作流体积；seam test 模式如何用文本断言锁定 GitHub Actions 结构；fail-closed 与 fork 回退如何防止静默降级。合并后建议先观察真实 PR 的门控时序再继续演进。

功能与动机

Miles CPU CI 频繁饱和（body 原文 'Miles CPU CI is frequently saturated'），而两个 CPU 阶段此前都要等 Docker 路径检测、可选镜像构建和镜像解析完成才启动，快速信号被拖慢。本重构把快速 CPU 信号提到最前（body 原文 'prioritizes the fast CPU signal'），同时保留测试选择、GPU 镜像行为以及 main 上新加入的 release 工作流。

实现拆解

1. 提取可复用工作流。新建 .github/workflows/_build-pr-ci-image.yml，将 docker-paths（改动检测）与 docker-build（多架构镜像构建）从 pr-test.yml 迁入，通过 workflow_call 暴露 built 输出，主工作流只保留调用与门控。
2. 调整主工作流依赖。stage-a-cpu 与 stage-b-cpu 的 needs 从 [resolve-ci-policy, resolve-ci-image] 改为 [resolve-ci-policy]；新增 docker-build caller，needs: [resolve-ci-policy, stage-a-cpu]，门控条件含 always()、事件非 closed、policy 成功、stage-a-cpu 成功或失败且 bypass_fastfail 为真。
3. 测试锁定契约。tests/ci/test/test_run_suite.py 的 TestWorkflowScopeSeam 将 test_both_cpu_stages_require_both_resolvers 改为 test_cpu_stages_only_require_policy，新增 test_docker_build_waits_for_cpu_gate_and_preserves_bypass 与 test_docker_build_body_lives_in_reusable_workflow，用文本断言锁定 caller 白名单、旁路分支与可复用文件驻留位置。
4. 配套文档与技能。docs/ci/00-stage.md 更新依赖表与门控说明，docs/ci/02-docker-build.md 改为指向可复用工作流入口；.claude/skills/doc-dev/SKILL.md 新增 base 差分减法 pass 规则。
5. 验证。pytest -q tests/ci/test 345 passed、1 skipped；pre-commit 与 git diff --check 通过；actionlint 未安装，未运行。

关键文件：

- .github/workflows/pr-test.yml（模块 主工作流；类别 infra；类型 infrastructure）：CI 主工作流：删除 85 行内联 docker 逻辑，两个 CPU 阶段脱离镜像解析，docker-build 改为受 stage-a-cpu 门控的可复用工作流调用。
- .github/workflows/_build-pr-ci-image.yml（模块 镜像构建；类别 infra；类型 infrastructure）：新增可复用工作流，承载 docker-paths 路径检测与 docker-build 镜像构建，暴露 built 输出，集中处理 fork、fail-closed 与 pip 引导。
- tests/ci/test/test_run_suite.py（模块 测试锁定；类别 test；类型 test-coverage；符号 test_both_cpu_stages_require_both_resolvers, test_cpu_stages_only_require_policy, test_docker_build_waits_for_cpu_gate_and_preserves_bypass, test_docker_build_body_lives_in_reusable_workflow）：用文本断言锁定 pr-test.yml 与可复用工作流的结构契约，防止后续改动破坏 CPU 门控与旁路语义。
- docs/ci/00-stage.md（模块 阶段文档；类别 docs；类型 documentation）：同步阶段依赖表与门控说明，记录 CPU 阶段脱离镜像解析、镜像链由 stage-a-cpu 门控的新契约。
- docs/ci/02-docker-build.md（模块 构建文档；类别 docs；类型 documentation）：把 docker-paths/docker-build 的说明改为指向可复用工作流 _build-pr-ci-image.yml 的入口。
- .claude/skills/doc-dev/SKILL.md（模块 文档技能；类别 docs；类型 documentation）：新增 base 差分减法 pass 规则，要求交付前删除非契约文案，保持文档改动最小化。

关键符号：test_cpu_stages_only_require_policy, test_docker_build_waits_for_cpu_gate_and_preserves_bypass, test_docker_build_body_lives_in_reusable_workflow

关键源码片段

.github/workflows/pr-test.yml

CI 主工作流：删除 85 行内联 docker 逻辑，两个 CPU 阶段脱离镜像解析，docker-build 改为受 stage-a-cpu 门控的可复用工作流调用。

```
  # 镜像构建入口：快速 CPU 信号（stage-a-cpu）通过后再进入，避免阻塞 CPU 反馈
  docker-build:
    needs: [resolve-ci-policy, stage-a-cpu]
    if: |
      always() && !cancelled() &&
      github.event.action != 'closed' &&
      needs.resolve-ci-policy.result == 'success' &&
      (needs.stage-a-cpu.result == 'success' ||
       (needs.stage-a-cpu.result == 'failure' && needs.resolve-ci-policy.outputs.bypass_fastfail ==
       'true'))
    uses: ./.github/workflows/_build-pr-ci-image.yml
    secrets: inherit

  # resolve-ci-image 消费可复用工作流的 built 输出，GPU 套件依赖它保持 fail-closed
  resolve-ci-image:
    needs: [docker-build]
```

```
    runs-on: ubuntu-latest
```

tests/ci/test/test_run_suite.py

用文本断言锁定 pr-test.yml 与可复用工作流的结构契约，防止后续改动破坏 CPU 门控与旁路语义。

```
  def test_docker_build_waits_for_cpu_gate_and_preserves_bypass(self):
      workflow = self._workflow()
      # 截取 docker-build caller 段，验证 CPU 门控与旁路契约
      caller = workflow.split("  docker-build:", 1)[1].split("  resolve-ci-image:", 1)[0]

      # 镜像构建必须等待 stage-a-cpu，且策略解析必须先成功
      assert "needs: [resolve-ci-policy, stage-a-cpu]" in caller
      # 即使上游被跳过或失败，也用 always() 让条件表达式统一评估
      assert "always() && !cancelled()" in caller
      # PR 关闭事件只做取消，不触发构建
      assert "github.event.action != 'closed'" in caller
      # stage-a-cpu 失败时，只有 bypass_fastfail 才能放行镜像链
      assert "needs.stage-a-cpu.result == 'failure'" in caller
      assert "needs.resolve-ci-policy.outputs.bypass_fastfail == 'true'" in caller
      # stage-b-cpu 保持并行，不能成为镜像链的前置
      assert "stage-b-cpu" not in caller
      # 调用可复用工作流并透传 secrets，保证镜像推送凭证可用
      assert "uses: ./.github/workflows/_build-pr-ci-image.yml" in caller
      assert "secrets: inherit" in caller
```

评论区精华

本 PR 无 review 评论，yushengsu-thu 直接 APPROVED（空 body）。PR body 的 Review Focus 提出了三个审核入口：pr-test.yml 的 stage-a-cpu 结果白名单与 release-cadence 旁路、_build-pr-ci-image.yml 的输出传播与 fork/fail-closed 行为、doc-dev 减法 pass 的误删风险；这些点最终由新增测试逐一覆盖，合并前没有遗留未解决疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 反馈时序变化：Docker 相关 PR 的镜像构建延后到 CPU 通过之后，GPU 套件启动随之变晚；这是有意的快速信号优先，但需要团队知晓。
  - 静态校验缺口：actionlint 未运行，YAML 拼写与表达式错误只能靠真实运行发现；当前由 seam 测试做文本级兜底。
  - fail-closed 语义：git diff 失败现在显式 ::error 并退出 1，会停住整条下游链；旧行为会把失败当空变更集静默跳过。
  - doc-dev 主观性：减法 pass 依赖执行者对删除后是否一致的判断，可能影响后续文档修改节奏。

- 影响范围：所有 PR 开发者的 CI 反馈顺序、CI 维护者的工作流结构、文档协作流程；对训练运行时无影响。
- 影响：对开发者：普通 PR 的 CPU 测试反馈更快（不再等镜像解析），但 Docker 相关 PR 的镜像与 GPU 反馈延后到 CPU 门控之后。对 CI 系统：pr-test.yml 减少约 85 行内联逻辑，Docker 细节收敛到可复用工作流，职责更清晰。对团队：CI 文档反映新契约，doc-dev 技能新增减法 pass，影响后续文档修改流程。兼容性方面，nightly、weekly、release cadence、MI350 覆盖、fork 行为与 bypass-fastfail 均保持。
- 风险标记：CI 核心流程重构，actionlint 未运行，Docker 构建反馈延后，fail-closed 语义变化，seam 测试兜底

关联脉络

- PR #2538 release: miles version release workflow: 同一时期加入 release 工作流，与 pr-test.yml 门控逻辑耦合；本 PR 明确保持 release cadence 与 release image/ref 输入行为不变。
- PR #2499 feat(ci): add weekly full-suite cadence: 引入 cadence 策略输出与调度门控，本 PR 的 stage-a-cpu 白名单和 bypass-fastfail 旁路直接消费 resolve-ci-policy 的输出。
- PR #2480 [AMD CI] Enable verified ROCm tests: 启用 MI350 ROCm 套件，共享 resolve-ci-policy/resolve-ci-image 逻辑；本 PR 声明 MI350 覆盖不变，后续改动需同步评估 pr-test-rocm.yml。