PR #2675 完整报告

radixark/miles

[CI] rebuild a PR's docker image only when its build inputs change

合并时间：2026-08-21 06:44

原文链接：http://prhub.com.cn/radixark/miles/pull/2675

执行摘要

- 一句话：CI 仅在实际镜像输入变化时重建 PR 镜像
- 推荐动作：建议精读。这是一个内容寻址式 CI 缓存决策的优秀范例：以输入哈希而非路径 diff 驱动重建、用一次性标签提供手动逃生舱、把注册表认证放到决策链末端。值得学习的设计决策包括：read_label 对 multi-arch / single-arch manifest 的递归兼容、用 HEAD^1 而非 base.sha 避免 base 滞后、live-label 读取规避 webhook payload 冻结问题。对同样维护昂贵 Docker 构建的团队有直接借鉴价值。

功能与动机

PR body 明确指出：A Dockerfile-touching PR previously rebuilt the same multi-arch image on every rerun or source-only push because CI keyed the build on changed paths. That repeated an expensive build even when the image inputs were unchanged。即旧的路径匹配策略把“触碰过 Docker 相关路径”当成“需要重建”，导致同一 PR 每次 rerun 或纯源码 push 都重复付出昂贵的多架构构建成本；本 PR 希望只在实际镜像输入（Dockerfile、requirements、patch 等）内容变化时才重建。

实现拆解

1. 新增输入哈希层 docker/image_inputs.py：定义 INPUT_GLOBS（docker/Dockerfile、docker/build.py、docker/install-kube-tools.sh、docker/verify_transformer_engine.py、docker/patch/*、requirements.txt），compute() 对工作区或指定 rev 下匹配文件的路径与内容做确定性 SHA-256 哈希；read_label() 递归解析 docker buildx imagetools inspect 的 JSON，兼容 multi-arch 与 single-arch manifest 的结构差异。刻意排除 Dockerfile.rocm，因为 ROCm 由独立流水线负责。
2. 构建时打标签 docker/build.py：在 build_and_push 中追加 --label miles.image-inputs=<hash>，让已发布的 pr-<num> tag 自带输入指纹，供后续决策回读——这是决策循环的写入端。
3. 重写决策 job _build-pr-ci-image.yml：原 docker-paths 改为 docker-decide，用 image_inputs.py --rev HEAD 与 --rev HEAD^1 比较 PR merge 提交与当前 base 的输入哈希；相等则声明 needs_image=false，非 Docker PR 或 fork PR 直接走 dev 镜像；需要镜像时通过 gh api 读取 PR 实时标签中的 rebuild-ci-image 决定是否强制重建，并将 tag_available 暴露给下游。

4. rerun 安全与标签消费：成功的强制构建会移除 rebuild-ci-image 标签（一次性语义），因此用 gh api --paginate repos/${GITHUB_REPOSITORY}/issues/${PR_NUMBER}/labels 读取实时状态而非冻结的 webhook payload；pr-test.yml 将 pull-requests: write 权限收窄到 docker-build job，仅该 job 消费标签。
5. 下游镜像选择 pr-test.yml：用 tag_available（本 run 构建或此前已发布）替代原来的 built，保证复用旧镜像的 run 仍用 pr-<num> 测试，而不是静默回退到正式发布镜像。
6. 测试与文档配套：新增 tests/fast/test_docker_image_inputs.py（globs 匹配、哈希确定性、输入追踪、标签解析），更新 tests/ci/test/test_run_suite.py 锁定 docker-decide 的步骤顺序与 live-label 用法，并同步 docs/ci/01-label.md、docs/ci/02-docker-build.md。

关键文件：

- docker/image_inputs.py（模块 镜像构建；类别 infra；类型 infrastructure；符号 compute, read_label, _matches, _paths_at）：新建的输入哈希模块，是整套"内容变化才重建"机制的单一事实来源：定义 INPUT_GLOBS、compute() 确定性哈希与 read_label() 标签回读。
- tests/fast/test_docker_image_inputs.py（模块 输入哈希；类别 test；类型 test-coverage；符号 test_dockerfile_and_requirements_are_inputs, test_rocm_dockerfile_is_not_a_cu13_input, test_source_changes_are_not_image_inputs, test_compute_is_deterministic）：新增 60 行测试覆盖输入 globs 边界（Dockerfile.rocm 排除、源码与文档不算输入）、哈希确定性、输入追踪与五种 manifest 的标签解析。
- tests/ci/test/test_run_suite.py（模块 CI 测试；类别 test；类型 test-coverage；符号 test_docker_decision_logs_in_only_for_tag_inspection, test_docker_force_rebuild_uses_live_label_for_decision_and_consumption）：更新 workflow 结构断言（docker-paths 改为 docker-decide），并新增两个关键不变量测试：login 必须晚于输入比较、强制重建必须读取实时 PR 标签。
- .github/workflows/_build-pr-ci-image.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：镜像构建决策的主体改动：新增 docker-decide job，承载输入哈希比较、实时标签读取、Docker Hub 登录顺序与 tag_available 输出。
- .github/workflows/pr-test.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：下游 run 用 tag_available 选择复用镜像，并把 pull-requests: write 权限收窄到 docker-build job 以支持标签消费。
- docker/build.py（模块 镜像构建；类别 infra；类型 infrastructure；符号 build_and_push）：构建命令追加 miles.image-inputs 标签，把输入哈希固化到已发布 tag 上，是决策循环的写入端。
- docs/ci/02-docker-build.md（模块 CI 文档；类别 docs；类型 documentation）：更新 Docker 构建流程文档，说明输入哈希决策与手动重建边界。
- docs/ci/01-label.md（模块 CI 文档；类别 docs；类型 documentation）：补充 rebuild-ci-image 标签说明。

关键符号：compute, read_label, _matches, _paths_at, _content_at, main, build_and_push

关键源码片段

tests/ci/test/test_run_suite.py

更新 workflow 结构断言（docker-paths 改为 docker-decide），并新增两个关键不变量测试：login 必须晚于输入比较、强制重建必须读取实时 PR 标签。

```
    def test_docker_decision_logs_in_only_for_tag_inspection(self):
        reusable = self._reusable_workflow("_build-pr-ci-image.yml")
        decide_job = reusable.split("  docker-decide:", 1)[1].split("  docker-build:", 1)[0]

        # 关键顺序不变量：先比较输入哈希，再登录 Docker Hub，最后 inspect 已发布 tag。
        # source-only PR 在比较阶段提前退出，完全不触碰 registry 凭据与可用性。
        compare = decide_job.index("python3 docker/image_inputs.py --rev HEAD^1")
        login = decide_job.index("- name: Login to Docker Hub")
        inspect = decide_job.index("docker buildx imagetools inspect")
        assert compare < login < inspect
        # 只有需要检查已发布 tag 时才执行 inspect
        assert "if: steps.prepare.outputs.inspect_tag == 'true'" in decide_job

    def test_docker_force_rebuild_uses_live_label_for_decision_and_consumption(self):
        reusable = self._reusable_workflow("_build-pr-ci-image.yml")
        decide_job = reusable.split("  docker-decide:", 1)[1].split("  docker-build:", 1)[0]

        # 必须查实时 PR 标签，而不是 webhook payload 里冻结的 labels：
        # Rerun all jobs 会复用同一 payload，冻结标签会让已消费的 force-rebuild
        再次触发全量构建。
        live_labels = '"repos/${GITHUB_REPOSITORY}/issues/${PR_NUMBER}/labels"'
        assert live_labels in decide_job
        assert decide_job.index(live_labels) < decide_job.index("- name: Login to Docker Hub")
        assert "LABELS=$(gh api --paginate" in decide_job
        assert 'grep -Fxq "rebuild-ci-image" <<< "$LABELS"' in decide_job
        # 决策与消费都依赖同一个实时标签，成功后移除标签实现一次性语义
        assert "needs.docker-decide.outputs.force_rebuild == 'true'" in reusable
```

评论区精华

guapisolo 在 merge 前提出两条 P2 评论，并均在最终 commit（34f7e6f1，由 guapisolo 本人提交）中解决。其一指出 github.event.pull_request.labels 冻结于 webhook payload：成功构建移除 rebuild-ci-image 后，Rerun all jobs 仍会读到 true 并再次全量重建，破坏一次性语义；因此改为在决策时点通过 gh api 查询实时 PR 标签，并让决策与消费共用同一份实时结果。其二指出原实现先登录 Docker Hub 再走 CURRENT == BASE 的早期退出路径，source-only 同仓 PR 也会依赖 registry 凭据与可用性；最终把输入比较移到认证之前，仅在需要 inspect 已发布 tag 时才登录。

- 强制重建需读取实时 PR 标签而非 webhook payload (correctness): 第三个 commit（34f7e6f1，guapisolo 提交）改为在决策时点用 gh api --paginate 查询 repos/{repo}/issues/{pr}/labels 的实时状态，决策与消费共用该结果，成功后移除标签实现一次性消费。

- Docker Hub 登录应推迟到真正需要镜像时 (design): 已采纳：docker-decide 中先比较输入哈希，再按需登录并 inspect 已发布 tag；inspect 步骤带 if 条件。配套测试 test_docker_decision_logs_in_only_for_tag_inspection 锁定该顺序。

风险与影响

- 风险：关键风险是输入清单的完备性：若日后 cu13 构建新增消费文件但未同步 INPUT_GLOBS，哈希不会变化，CI 会静默复用旧镜像；测试 test_compute_tracks_every_declared_input 只能防止测试层面的人为篡改，无法防止真实遗漏；Dockerfile.rocm、base 镜像漂移、floating 依赖等属于显式手动重建边界，需要团队维护心智。其次，live-label 决策依赖 gh api 的可用性与 token 权限（pull-requests: write 已收窄到 docker-build job），GitHub API 限流或 GHES 环境差异可能让强制重建路径失效。再次，--rev HEAD 与 --rev HEAD^1 依赖 GitHub Actions 生成的 merge commit 结构，若未来 checkout 策略变化（fetch-depth、merge 行为）需要同步调整。最后，镜像复用意味着测试结果在被复用的镜像上产生，若镜像已损坏但哈希未变，需要人工通过 rebuild-ci-image 标签介入。
- 影响：对用户侧无直接功能影响；对仓库 CI 影响显著：所有 Docker 相关 PR 的 rerun 与纯源码 push 将复用已发布的 pr-<num> 镜像，省去每次昂贵的多架构构建；非 Docker PR 与 fork PR 不再触碰 Docker Hub 认证，降低 registry 故障对 GPU 流水线的连带影响。团队需要新增一个运维知识：在 base 镜像漂移、floating 依赖升级或镜像损坏时给 PR 打 rebuild-ci-image 标签触发强制重建，成功后标签会被消费移除。对 workflow 维护者来说，docker-decide 成为镜像决策的唯一入口，后续任何构建输入的变更都应同步到 INPUT_GLOBS。
- 风险标记：输入清单遗漏致镜像静默过期，GitHub API 限流影响强制重建，merge commit 结构假设，镜像复用依赖人工逃生舱

关联脉络

- PR #2496 feat(ci): add authorized comment-to-label gateway: 该 PR 建立 PR 评论加 CI 标签的授权网关，本 PR 的 rebuild-ci-image 一次性标签依赖同一标签基础设施，是其消费机制的第一个用例。
- PR #2671 [AMD] Enable four CI tests on ROCm: 本 PR 刻意将 Dockerfile.rocm 排除在 cu13 输入之外，因为 ROCm 镜像由独立流水线维护；与 2671 的 ROCm CI 配置边界互为印证。
- PR #2600 fix(docker): pin cutlass-dsl 4.6.2 and flashinfer 0.6.15.post1 over the sglang base: base 镜像漂移与 floating 依赖升级正是 PR body 中列为需要 rebuild-ci-image 手动重建的典型场景。
- PR #2670 fix(docker): update torch_memory_saver for CUDA VMM granularity: 同为 Docker 依赖变更触发重建的实例；本 PR 之后这类依赖升级若未改动输入文件，需靠 rebuild-ci-image 标签强制重建。