

PR #2363 完整报告

radixark/miles

ci(docker): run image builds on docker-build runners

合并时间: 2026-08-11 15:09

原文链接: <http://prhub.com.cn/radixark/miles/pull/2363>

执行摘要

- 一句话: Docker 镜像构建迁移至专用 docker-build runner
- 推荐动作: 值得快速阅读, 作为 CI 资源优化的小而典型案例。重点关注两点: 一是用实测数据 (72 分钟 GPU 池占用) 驱动调度决策的方法; 二是通过注释明确“行为保持”边界, 让 reviewer 只需确认 runner 选择逻辑即可放心合并。若后续要扩展 docker-build 节点规模, 可参考此 PR 的契约描述。

功能与动机

PR body 明确说明: 一次自动 Docker 构建占用了 ["h200", "2gpu"] 池长达 72 分钟, 即使 buildx-sandboxed 的 RUN 步骤并未使用 GPU。这种占用消耗了 GPU-suite 容量却对镜像产物没有任何改善, 因此将镜像构建路由到带 docker-build 标签的专用 runner 实例, 释放 GPU 资源给真正的 GPU 任务。

实现拆解

1. .github/workflows/docker-build.yml: build-and-push job 的 runs-on 从 ["h200", "2gpu"] 改为 ["docker-build"], 并新增注释说明 buildx 会沙箱化 RUN 步骤、docker-build 节点是携带第二个 runner 实例的 CI 机器, 从而让镜像构建不再与 GPU-suite 竞争 runner 槽位。
2. .github/workflows/pr-test.yml: docker-build job 保留原有条件表达式 (Docker 路径有变更且 PR 来自本仓库), 满足条件时 runs-on 从 GPU 池切换到 ["docker-build"], 否则仍然回落到 ubuntu-latest 托管 runner 执行 no-op; 同时更新了顶部注释, 强调只有真实构建才占用专用节点。
3. docs/ci/02-docker-build.md: 将 build-and-push 的 runner 描述从 self-hosted runner 更新为 self-hosted docker-build runner, 保证文档与新的调度契约一致。
4. 第二个 commit 将 main 合并进分支, 解决 pr-test workflow 的冲突, 同时保留 closed-PR 取消逻辑 (github.event.action != 'closed')。验证方式为 git diff --check、PyYAML BaseLoader 解析两个 workflow, 以及对 live main 的三点 diff 确认无 Dockerfile 或构建脚本改动。

关键文件:

- .github/workflows/pr-test.yml (模块 PR 流水线; 类别 infra; 类型 infrastructure): PR 镜像构建的 runner 选择逻辑所在, 从 GPU 池转向 docker-build 节点的关键一处, 且保留

了 fork/no-op 回退路径。

- `.github/workflows/docker-build.yml`（模块 镜像构建；类别 infra；类型 infrastructure）：定时 / 上游触发镜像构建的主入口，全面迁移到 docker-build runner，并新增注释说明专用节点机制。
- `docs/ci/02-docker-build.md`（模块 CI 文档；类别 docs；类型 documentation）：同步 CI 文档中的 runner 契约，将 build-and-push 的 runner 描述更新为 docker-build，避免文档与调度规则脱节。

关键符号：未识别

关键源码片段

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

PR 镜像构建的 runner 选择逻辑所在，从 GPU 池转向 docker-build 节点的关键一处，且保留了 fork/no-op 回退路径。

```
# docker-build job——仅在真实需要构建时使用专用节点：
# 1. docker-paths 检测到 Docker 相关路径变更；
# 2. PR 来源仓库是 radixark/miles（非 fork），fork 只做 no-op 检查。
# 满足条件时在 docker-build 专用 runner 上跑镜像构建，
# 否则落到 ubuntu-latest 托管 runner 上执行空步骤。
docker-build:
  needs: [docker-paths]
  if: always() && !cancelled() && github.event.action != 'closed'
  runs-on: ${ fromJSON((needs.docker-paths.outputs.changed == 'true' && github.event.pull_
    request.head.repo.full_name == github.repository) && "[\"docker-build\"]" || "[\"ubuntu-latest\"]") }
  timeout-minutes: 180
  outputs:
    built: ${ steps.build.outputs.built || 'false' }
```

评论区精华

该 PR 没有产生任何 review 评论或讨论线程；yueming-yuan 与 yushengsu-thu 两位维护者均直接 APPROVED，说明变更无争议。可见的决策要点全部沉淀在代码注释中：buildx 沙箱使得构建阶段不接触 GPU，因此可以安全迁移到专用节点；docker-build 节点通过第二个 runner 实例承载构建，避免与 GPU 测试抢资源。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于对 docker-build 专用 runner 可用性与容量的依赖。若该节点离线或数量不足，镜像构建将排队或失败，且没有自动回退到 GPU 池的路径，需要人工介入扩容。其次，timeout-minutes 仍保持 180 分钟，若专用节点资源配置较弱，长构建可能更接近超时边界。对 fork PR 与 no-op 路径行为完全不变，安全边界没有被破坏；该变更为纯 CI 调度调整，不影响训练 / 推理运行时。

- 影响：影响集中在 CI 基础设施：Docker 镜像自动构建（含定时触发的 build-and-push 与 PR 触发的 docker-build job）不再占用 H200 双卡 GPU runner，GPU-suite 的排队等待时间应有所改善。对用户侧无感知，镜像产物、标签和构建逻辑完全一致；对团队而言是一次低风险、低成本的基础设施优化，并同步更新了 CI 文档以保持信息一致。
- 风险标记：docker-build runner 可用性依赖，无自动回退路径，仅调度变更，无新增测试覆盖

关联脉络

- PR #2290 fix(ci): persist every step metric for historical gate: 同为 CI 基础设施演进，完善 CI 指标与门禁，与此 PR 优化 CI 资源分配属于同一维护方向。
- PR #2279 Run the launch script snapshot tests by hand instead of in CI: 将非必要测试工作移出 CI，与此 PR 将非 GPU 构建移出 GPU 池的目标一致，共同降低 CI 资源占用。