

PR #28853 完整报告

sgl-project/sglang

[AMD CI] Add nightly Miles ROCm 7.2 MI350X suites

合并时间: 2026-06-26 18:47

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28853>

执行摘要

此 PR 为 sglang 仓库添加了一个针对 AMD MI350X (ROCm 7.2) 硬件的夜间 CI 工作流。它通过自动解析最新 dated miles 镜像, 运行三套 e2e 测试套件 (2/4/8-GPU), 填补了 AMD 硬件回归测试的空白。无需修改核心 Python/CUDA 代码。

功能与动机

PR body 明确说明: “Add a nightly GitHub Actions workflow for the Miles ROCm 7.2 MI350X e2e suites.” 动机是为 AMD MI350X 硬件在代码变更后提供持续的自动化回归检测, 提升平台稳定性。

实现拆解

1. 工作流定义: 新增 `.github/workflows/nightly-test-amd-miles-rocm720.yml`, 配置 cron 调度 (UTC 17:30) 和手动触发 (`workflow_dispatch`), 支持自定义镜像标签和继续运行出错开关。
2. 并发与矩阵策略: 使用 `concurrency` 组避免重复执行; 通过矩阵定义三个套件 (`stage-c-2-gpu-mi350`、`stage-c-4-gpu-mi350`、`stage-c-8-gpu-mi350`), 分别映射到自托管 runner: 2-GPU 套件使用 `linux-mi35x-gpu-2`, 其余使用 `linux-mi35x-gpu-8`。
3. 镜像解析: 首先登录 Docker Hub, 然后遍历最近 7 天日期, 用 `docker manifest inspect` 探测 `rocm/sgl-dev:miles-rocm720-mi35x-<date>` 标签是否存在, 避免实际拉取镜像。若无可用标签则报错退出。
4. 环境准备: 调用 `ensure_vram_clear.sh` 清理显存, 然后通过 `amd_ci_start_container.sh` 启动 miles 容器, 并传递自定义镜像变量 `MILES_IMAGE`。
5. 测试执行: 在容器内设置 `PYTHONPATH=/root/miles:/opt/tilelang`, 然后运行 `python3 tests/ci/run_suite.py --hw rocm --suite <suite> --match-all-labels`, 支持 `--continue-on-error` 参数。测试结果写入 GitHub Step Summary。

此 PR 为纯 CI 配置变更 (YAML), 无核心函数或类变更, 因此不展示源码片段。详细配置可查看 [nightly-test-amd-miles-rocm720.yml](#)。

评论区精华

- `PYTHONPATH` 覆盖: @bingxche 指出原始覆盖会丢失 `/opt/tilelang`。@XinyuJiangCMU 随即修复为包含两个路径。

- 镜像解析来源: @bingxche 坚持使用公共 Docker Hub 而非本地 registry, @XinyuJiangCMU 修改为 docker manifest inspect 探测公共标签。
- 外部依赖: @bingxche 提问 --hw rocm 是否依赖 miles#1379, @XinyuJiangCMU 确认是, 当前提交可能暂时不可用。

风险与影响

- 镜像可用性: 若 dated 镜像未按时推送或超过 7 天, workflow 报错退出。
- 环境差异: 4-GPU 套件在 8-GPU runner 上执行, 可能掩盖特定连通性问题。
- 外部依赖: 测试套件注册在 miles 仓库, sglang 无法控制其注册时机, 可能导致空套件。
- 资源占用: 多矩阵任务可能长时间占用 GPU runner。

影响范围局限于 AMD CI 基础设施, 对用户无感知, 但可提升 AMD 硬件的测试覆盖和发布信心。

关联脉络

此 PR 与 #29084 形成 AMD nightly 测试体系: 前者覆盖 MI355X 分解基准测试, 后者覆盖 MI350X 通用 e2e 测试。共同体现了 sglang 社区对 AMD 硬件持续集成投入的增加。