

PR #2480 完整报告

radixark/miles

[AMD CI] Enable verified ROCm tests

合并时间: 2026-08-15 04:05

原文链接: <http://prhub.com.cn/radixark/miles/pull/2480>

执行摘要

- 一句话: 启用 21 个已验证测试至 MI350 ROCm 套件
- 推荐动作: 值得快速浏览, 尤其是 tests/ci/run_suite.py 的套件拆分配置和任意一个测试文件的 register_rocm_ci 注册模板。该 PR 展示了如何在保持测试体不变的前提下, 将已验证的测试安全接入新硬件 CI 通道, 是后续新增 ROCm 测试的参考样板。若团队后续要继续扩 ROCm 覆盖, 建议关注 est_time 校准是否持续基于实测数据, 以及外部 sgl-project/sglang nightly 对新套件名的消费是否稳定。

功能与动机

PR body 明确说明动机是“Enable 21 existing CI tests on the MI350 ROCm suites. Their test bodies pass on MI355X, so this PR only changes ROCm CI registrations.”即在 AMD 侧建立可信的 nightly 回归覆盖, 让已经在 8x MI355X 上验证通过的测试 (镜像 rocm/sgl-dev:miles-rocm720-mi35x-20260811) 进入自动化套件, 避免 ROCm 功能缺乏持续回归保护。

实现拆解

实现拆解如下:

1. 补充 ROCm 注册: 在 36 个测试文件中将 from tests.ci.ci_register import register_cuda_ci 扩展为同时导入 register_rocm_ci, 并在原有 CUDA 注册之后追加 register_rocm_ci(est_time=..., suite="nightly-stage-* gpu-mi350", labels=[...]) 调用。典型例子是 tests/e2e/fsdp/test_qwen3_0.6B_fsdp_distributed.py, 在 CUDA 的 stage-c-4-gpu-h200 之外新增长标签的 nightly-stage-c-4-gpu-mi350 注册。
2. 统一 nightly- 套件前缀: 提交 fa45c98 将本 PR 新增的活动 ROCm 注册统一改为 nightly- 前缀, 例如 tests/e2e/megatron/test_qwen3_30B_A3B/test_baseline.py 从 stage-c-4-gpu-mi350 改为 nightly-stage-c-4-gpu-mi350, 用于与 PR 级 CI 套件 (stage-c-4-gpu-mi350, 由 pr-test-rocm.yml 消费) 和外部 sgl-project/sglang MI350 nightly 消费的套件区分。
3. 校准耗时估计: 提交 f527737 根据 MI355X 实测运行时长校准各测试的 est_time, 例如 test_qwen3_0.6B_fsdp_colocated_2xGPU.py 的 ROCm 注册为 3900 秒 (CUDA 侧 3000 秒), test_qwen3_4B_ppo.py 为 800 秒, test_qwen3_4B_offload_disk.py 为 300 秒, test_run_megatron_worker_main.py 为 20 秒。

4. 更新套件列表与文档: tests/ci/run_suite.py 中 HWBackend.ROCM 的可用套件列表改为 stage-c-4-gpu-mi350 (PR CI 消费) 加 nightly-stage-c-8/4/2-gpu-mi350 (外部 nightly 消费); 提交 373cde 同步更新了 CI 文档, 说明 nightly- 前缀语义并列出 MI350 套件。
5. 格式修复与合并: 最后一笔提交 c5488c1 处理了 black 报出的两处超行 register_rocm_ci 调用格式问题, 随后合并入 main。

测试与配置配套: 所有变更都是测试注册与 CI 配置层面, 没有修改任何测试体、断言、运行参数或正确性阈值; `register_ci_gate` 的指标门槛保持不变。

关键文件:

- tests/ci/run_suite.py (模块 套件配置; 类别 test; 类型 configuration; 符号 HWBackend.ROCM): 核心 CI 配置变更: 调整 HWBackend.ROCM 可用套件列表, 区分 PR CI 消费的 stage-c-4-gpu-mi350 与外部 sgl-project/sglang nightly 消费的 nightly-* 套件, 决定 ROCm 测试实际调度目标。
- tests/e2e/fsdp/test_qwen3_0.6B_fsdp_distributed.py (模块 分布式测试; 类别 test; 类型 test-coverage): FSDP 分布式训练测试的 ROCm 注册模板, 4 GPU MI350 套件, 展示了本 PR 最典型的注册模式。
- tests/e2e/megatron/test_qwen3_4B_ppo.py (模块 PPO 测试; 类别 test; 类型 test-coverage): Megatron 4B PPO 训练测试, ROCm 注册 est_time 800 秒, 代表 PPO 训练路径的 ROCm 回归覆盖。
- tests/e2e/megatron/test_qwen3_4B_offload_disk.py (模块 磁盘卸载; 类别 test; 类型 test-coverage): 磁盘 offload 路径测试, 自带对“每个 rank 是否真正启用了 disk offload”的日志断言, ROCm 注册同时保留了 metric gate, 是断言较强的测试之一。
- tests/e2e/fsdp/test_qwen3_0.6B_megatron_fsdp_align.py (模块 对齐测试; 类别 test; 类型 test-coverage): FSDP 与 Megatron 对齐测试, 2 GPU MI350 套件, 验证两种训练后端对齐的 ROCm 回归。
- tests/e2e/short/test_qwen3_0.6B_fsdp_colocated_2xGPU.py (模块 共置测试; 类别 test; 类型 test-coverage): 2xGPU 共置 colocated 模式测试, ROCm est_time 3900 秒明显高于 CUDA 侧 3000 秒, 是本 PR 校准耗时估计的代表。
- tests/e2e/megatron/test_qwen3_30B_A3B/test_fully_async.py (模块 异步测试; 类别 test; 类型 test-coverage): fully-async 训练路径的 ROCm 注册, 带有 fully-async 标签, 8 GPU MI350 套件, 扩大异步训练在 AMD 侧的覆盖。
- tests/e2e/megatron/test_qwen3_30B_A3B/test_baseline.py (模块 基线测试; 类别 test; 类型 test-coverage): 展示既有 ROCm 注册从 stage-c-4-gpu-mi350 迁移到 nightly-stage-c-4-gpu-mi350 的过程, 是 nightly- 前缀规范化改造的示例。

关键符号: register_rocm_ci, register_cuda_ci, register_ci_gate

关键源码片段

tests/ci/run_suite.py

核心 CI 配置变更: 调整 HWBackend.ROCM 可用套件列表, 区分 PR CI 消费的 stage-c-4-gpu-mi350 与外部 sgl-project/sglang nightly 消费的 nightly-* 套件, 决定

ROCm 测试实际调度目标。

```
# tests/ci/run_suite.py —— ROCm 可用套件列表
# 拆分 PR CI 与外部 nightly 消费方，避免把 ROCm nightly 测试误跑进 PR 门槛
HWBackend.ROCM: [
    # 该套件被 pr-test-rocm.yml 消费，作为 AMD PR CI 的回归集合
    "stage-c-4-gpu-mi350",
    # 以下套件被外部 sgl-project/sglang 的 MI350 nightly 消费，
    # 通过 nightly- 前缀与 PR 级套件明确区分
    "nightly-stage-c-8-gpu-mi350",
    "nightly-stage-c-4-gpu-mi350",
    "nightly-stage-c-2-gpu-mi350",
]
```

tests/e2e/fsdp/test_qwen3_0.6B_fsdp_distributed.py

FSDP 分布式训练测试的 ROCm 注册模板，4 GPU MI350 套件，展示了本 PR 最典型的注册模式。

```
# tests/e2e/fsdp/test_qwen3_0.6B_fsdp_distributed.py
from tests.ci.ci_register import register_cuda_ci, register_rocm_ci

# CUDA 侧注册保持不变
register_cuda_ci(
    est_time=2200,
    suite="stage-c-4-gpu-h200",
    labels=["long"],
)
# 新增 ROCm 注册：进入 MI350 nightly 套件，耗时按 MI355X 实测校准为 2400 秒
register_rocm_ci(
    est_time=2400,
    suite="nightly-stage-c-4-gpu-mi350",
    labels=["long"],
)
```

评论区精华

本 PR 的 review 结论非常简洁：审核人 [guapisolo](#) 给出了 APPROVED，但附带一句“approve. but need handle pre commit issues.”，指出合并前需要处理 pre-commit 报出的格式问题。该问题在最后一个提交 [c5488c1](#) 中得到解决：[nightly-](#) 前缀使两处 [register_rocm_ci](#) 调用超过行宽，作者用 black 重新换行，且明确说明“No arguments changed.”。除此之外没有其他讨论线程或争议。

- pre-commit 格式问题 (other): 最后一个提交 [c5488c1](#) 明确说明 black 将两处因 [nightly-](#) 前缀而超行的 [register_rocm_ci](#) 调用重新换行，且 No arguments changed，问题已解决。

风险与影响

- 风险：主要风险集中在 CI 配置与调度层面：

- nightly 资源成本增加：新增 21 个测试进入 2/4/8 GPU MI350 套件，会显著增加 nightly 阶段的 GPU 占用；`est_time` 仅基于一次 8x MI355X 实测，MI350 与 MI355X 在驱动、ROCm 栈和硬件细节上的差异可能导致超时或误报。
- 套件命名依赖外部消费方：tests/ci/run_suite.py 将 nightly-stage-c-8/4/2-gpu-mi350 标注为“Consumed by the external sgl-project/sglang MI350 nightly”，若外部消费方没有同步识别新的 nightly- 前缀套件，会导致这些测试静默不运行或跑错集合。
- 注册分散、易漏检：36 个文件分散添加注册，若某个 suite 名拼写错误或 CI 策略校验未覆盖，测试可能被静默跳过；本次没有新增针对注册本身的校验测试。
- 正确性风险低：测试体与断言未改动，因此不会引入新的模型或训练逻辑回归风险。
- 影响：对用户与团队的影响：ROCm 用户将获得更全面的 nightly 回归覆盖，涉及 FSDP 分布式训练、Megatron PPO、磁盘 offload、fully-async、sglang session 等关键路径，有助于提前发现 AMD 后端的兼容性问题。对系统而言，nightly MI350 套件新增大量测试，需要持续提供 2/4/8 GPU 资源。对 CI 维护者而言，nightly- 前缀成为区分 PR 级与外部门禁消费方的事实约定，后续新增 ROCm 测试都应遵循此模式。CUDA 侧完全不受影响。
- 风险标记：nightly 资源成本增加，`est_time` 基于单次环境测量，套件命名依赖外部消费方，注册分散易漏检

关联脉络

- PR #2347 [AMD] Enable amd pr ci: 同一 AMD CI 建设主线：该 PR 启用 ROCm PR CI 通道与 AMD 训练脚本，本 PR 在其基础上把已验证测试注册进 nightly MI350 套件。
- PR #2499 feat(ci): add weekly full-suite cadence: 涉及 tests/ci/run_suite.py、ci_policy.py 等套件调度逻辑，本 PR 修改了同一文件中的 ROCm 套件列表，两者在套件选择策略上游存在依赖。
- PR #2397 fix(fsdps): move the AMD Triton attention bridge in-tree: 修复 ROCm FSDP 后端不可用问题的前置修复，本 PR 启用的 FSDP 相关 ROCm 测试（如 fsdp_distributed、megatron_fsdps_align）可为该修复提供 nightly 回归验证。