PR #2347 完整报告

radixark/miles

[AMD] Enable amd pr ci

合并时间：2026-08-14 11:14

原文链接：http://prhub.com.cn/radixark/miles/pull/2347

执行摘要

- 一句话：启用 ROCm PR CI 通道，新增 AMD 专属训练脚本
- 推荐动作：值得精读。该 PR 展示了在不污染 NVIDIA 共享路径的前提下引入硬件专属配方的完整套路：环境变量分发、后端中性参数透传、测试注册套件迁移。Review 中关于 Megatron 专家网格整除约束的讨论是可复用的并行配置知识，agentic 改动的拦截也体现了维护者对 PR 主题边界的把握。建议重点阅读 scripts/amd/run_inkling.py 的 _get_parallel_config 与 test_qwen3_30B_A3B/_common.py 的 CaseConfig 扩展。

功能与动机

PR body 明确说明要“Bring up the ROCm CI lane on MI300X/MI355X and enable the cases, without changing shared NVIDIA behavior”，即在 AMD 平台上获得与 NVIDIA 同等的 CI 回归能力。此前这些用例在 register_rocm_ci 中统一标记为 “Disable due to failure”，没有任何实际覆盖。关联 issue #2265 进一步说明 ROCm PyTorch 缺少私有 DeviceMesh._unflatten 导致 FSDP hybrid-shard 无法工作，需要通过 capability-based fallback 才能跑通。

实现拆解

1. 新增 AMD 专属训练配方目录：在 scripts/amd/ 下新增 run_glm5_2_744b_a40b.py、run_inkling.py（以及测试引用的 run_deepseek_v4.py），完整封装下载、checkpoint 校验、FP8 转换、Megatron checkpoint 转换与训练执行流程，保持与 scripts/ 下共享脚本同构，避免在 NVIDIA 路径中插入 ROCm 分支。
2. 测试层后端分发：各 e2e 测试文件（如 test_glm5_2_744b_a40b_5layer_ci.py、test_inkling_small_4layer_ci.py、test_deepseek_v4_flash_4layer_ci.py）在 import 阶段通过 os.getenv("MILES_HARDWARE_PLATFORM") == "rocm" 选择 scripts.amd.* 或共享模块；同时将 ROCm 注册套件从 stage-c-4-gpu-mi300x 统一迁移到 stage-c-4-gpu-mi350，并使用 disabled="FIXME: re-enable once this case passes on the MI350 runners." 保留未通过用例。
3. CaseConfig 后端中性扩展：在 tests/e2e/megatron/test_qwen3_30B_A3B/_common.py 的 CaseConfig 中新增 extra_args 与 extra_env_vars 字段，并在 build_train_args / execute 中注入训练命令与环境变量，用于传递 mori a2a backend、SGLANG_MORI_NUM_MAX_DISPATCH_TOKENS_PER_RANK 等 AMD 特有参数。

4. CI 基础设施调整：.github/workflows/pr-test-rocm.yml 与 _run-ci-rocm.yml 切换到 MI350 runner、rocm720 镜像与显式硬件选择器；tests/ci/labels.py、run_suite.py、tests/ci/test/test_run_suite.py 同步适配；docs/ci/00-stage.md 更新 run-ci-amd 标签的选择语义说明。
5. 附带改动：调整 miles/rollout/generate_hub/agentic_tool_call.py 中 use_v2 共享 rollout_id 的分配条件（仅多 leaf 时分配），该改动在 review 中被要求交由后续 PR 处理，合并前由 guapisolo 更新定稿。

关键文件：

- scripts/amd/run_glm5_2_744b_a40b.py（模块 AMD 脚本；类别 source；类型 core-logic；符号 ScriptArgs, post_init, _validate_glm_checkpoint, _convert_to_fp8）：新增的 AMD 专属 GLM-5.2 训练脚本，是本 PR 三大配方之一，完整承载 ROCm 下载、校验、FP8 转换与训练流程，是理解 AMD CI 接入方式的核心入口。
- scripts/amd/run_inkling.py（模块 AMD 脚本；类别 source；类型 core-logic；符号 ScriptArgs, post_init, _get_parallel_config, _train）：新增的 AMD 专属 Inkling 训练脚本，内置已验证的并行配置选择函数 _get_parallel_config，是 ROI 配方分发的另一个核心示例。
- tests/e2e/megatron/model_scripts/test_glm5_2_744b_a40b_5layer_ci.py（模块 GLM5 CI；类别 test；类型 test-coverage）：展示了本 PR 核心的"后端分发"模式：测试在 import 阶段根据 MILES_HARDWARE_PLATFORM 选择 scripts.amd.* 还是共享模块，并将 ROCm 套件迁移到 MI350 且标记 FIXME 禁用。
- tests/e2e/megatron/test_qwen3_30B_A3B/_common.py（模块 公共配置；类别 test；类型 test-coverage；符号 CaseConfig, build_train_args, execute）：CaseConfig 增加后端中性的 extra_args / extra_env_vars，是"小差异透传"模式的载体，被多个 AMD e2e 用例复用。
- .github/workflows/pr-test-rocm.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：ROCm PR CI 的执行入口，切换 MI350 runner、rocm720 镜像与显式硬件选择器，是本 PR 在基础设施层的落点。
- miles/rollout/generate_hub/agentic_tool_call.py（模块 生成中心；类别 source；类型 core-logic；符号 generate）：Review 中唯一被维护者明确反对的修改点，涉及 session v2 共享 rollout_id 语义，最终由 guapisolo 在合并前更新定稿，是理解本 PR 边界的重要文件。

关键符号：ScriptArgs.post_init, _execute_train, full_train, _train, _get_parallel_config, CaseConfig, build_train_args, execute, generate

关键源码片段

scripts/amd/run_glm5_2_744b_a40b.py

新增的 AMD 专属 GLM-5.2 训练脚本，是本 PR 三大配方之一，完整承载 ROCm 下载、校验、FP8 转换与训练流程，是理解 AMD CI 接入方式的核心入口。

```
# scripts/amd/run_glm5_2_744b_a40b.py（新增，ROCm 专用配方）

def _execute_train(args: ScriptArgs):
```

```
"""组装并执行 GLM-5.2 训练命令，AMD 特有的 SGLang 后端参数集中在这里。"""
load_save_path = f"{args.output_dir}/{args.run_id}/checkpoints"
hf_name = f"{args.model_name}_fp8" if args.fp8_rollout else args.model_name

ckpt_args = (
    f"--hf-checkpoint {args.model_local_dir}/{hf_name} "
    f"--ref-load {args.model_local_dir}/{args.model_name}_torch_dist "
    f"--load {load_save_path} "
    f"--save {load_save_path} "
    "--save-interval 20 "
)

# sglang 侧参数：mem-fraction 0.70 是 4xH200 冒烟测试验证过的值，
# 裁剪后的权重使 0.85 几乎全部变成 KV cache，会让 weight-checker 快照无处分配
sglang_args = (
    f"--rollout-num-gpus-per-engine {sglang_world_size} "
    "--sglang-mem-fraction-static 0.70 "
    f"--sglang-ep-size {sglang_world_size} "
    "--sglang-router-policy consistent_hashing "
)
if args.fp8_rollout and args.use_deepep:
    # ROCm 上 FP8 rollout 走 mori a2a 后端，而不是 CUDA 的 DeepEP
    sglang_args += "--sglang-moe-a2a-backend mori " "--sglang-deepep-mode auto "
sglang_args += (
    "--sglang-kv-cache-dtype fp8_e4m3 "
    "--sglang-nsa-decode-backend tilelang "
    "--sglang-attention-backend nsa "
)

# 将各段参数拼接，最后统一追加调用方传入的 extra_args，
# 使 CI 用例可以在不改脚本的情况下叠加 AMD 特有开关
train_args = (
    f"{ckpt_args} {rollout_args} {optimizer_args} {grpo_args} "
    f"{U.get_default_wandb_args(__file__, run_id=args.run_id)} "
    f"{perf_args} {sglang_args} {misc_args} "
    f"{args.extra_args} "
)
U.execute_train(
    train_args=train_args,
    config=args,
    num_gpus_per_node=args.num_gpus_per_node,
    megatron_model_type=args.megatron_model_type,
    extra_env_vars={
        "SGLANG_NSA_FORCE_MLA": "1",
        "INDEXER_ROPE_NEOX_STYLE": "0",
    },
    megatron_path=args.megatron_path,
)
```

```
@app.command()
@U.dataclass_cli
def full_train(args: ScriptArgs):
    """完整流水线：下载 -> 校验 -> FP8 转换 -> Megatron 转换 -> 训练。"""
    _prepare_download(args)
    _validate_glm_checkpoint(args)
    if args.fp8_rollout:
        _convert_to_fp8(args)
    _prepare_megatron_ckpt(args)
    _execute_train(args)
```

scripts/amd/run_inkling.py

新增的 AMD 专属 Inkling 训练脚本，内置已验证的并行配置选择函数 _get_parallel_config，是 ROI 配方分发的另一个核心示例。

```
# scripts/amd/run_inkling.py（新增，ROCm 专用配方）

@dataclass
class ScriptArgs(U.ExecuteTrainConfig):
    run_id: str = U.create_run_id()
    model_name: Literal["Inkling-Small-4layer"] = "Inkling-Small-4layer"
    hf_checkpoint: str | None = None
    torch_dist: str | None = None
    torch_dist_local: str | None = None
    enable_r3: bool = True
    extra_args: str = ""

    def __post_init__(self):
        if self.model_name.endswith("-4layer"):
            # 4 层裁剪复用基础模型定义，通过环境变量覆盖层数
            os.environ["MODEL_ARGS_NUM_LAYERS"] = "4"
        if self.hf_checkpoint is None:
            self.hf_checkpoint = f"{self.model_dir}/{self.model_name}"
        if self.torch_dist is None:
            self.torch_dist = f"{self.model_dir}/{self.model_name}_torch_dist"
        if self.torch_dist_local is None:
            self.torch_dist_local = self.torch_dist
        if self.lr is None:
            self.lr = 1e-6
        self.colocate = True
        self.actor_num_nodes = self.num_nodes
        self.actor_num_gpus_per_node = self.num_gpus_per_node


def _get_parallel_config(args: ScriptArgs) -> str:
    """返回经过验证的并行配置；未验证的拓扑直接抛 NotImplementedError。"""
    total_gpus = args.actor_num_nodes * args.actor_num_gpus_per_node

    # 4 卡单节点：TP4 + SP + EP4，所有 expert 分布到 4 张卡上
```

```
    if args.model_name == "Inkling-Small-4layer" and args.actor_num_nodes == 1:
        return (
            "--tensor-model-parallel-size 4 "
            "--sequence-parallel "
            "--pipeline-model-parallel-size 1 "
            "--expert-model-parallel-size 4 "
            "--expert-tensor-parallel-size 1 "
        )

    raise NotImplementedError(
        f"No pre-set parallel config for {total_gpus} GPUs. "
        f"Please specify your parallel config in `scripts/amd/run_inkling._get_parallel_config`."
    )
```

tests/e2e/megatron/model_scripts/test_glm5_2_744b_a40b_5layer_ci.py

展示了本 PR 核心的“后端分发”模式：测试在 import 阶段根据 MILES_HARDWARE_PLATFORM 选择 scripts.amd.* 还是共享模块，并将 ROCm 套件迁移到 MI350 且标记 FIXME 禁用。

```
# tests/e2e/megatron/model_scripts/test_glm5_2_744b_a40b_5layer_ci.py（核心分发模式）

import os

# 同一个测试文件通过环境变量选择不同平台的训练配方：
# ROCm 环境（由 Dockerfile.rocm 注入 MILES_HARDWARE_PLATFORM=rocm）
# 走 scripts.amd.*，其他平台走共享的 scripts.*，NVIDIA 路径不受影响
if os.getenv("MILES_HARDWARE_PLATFORM") == "rocm":
    from scripts.amd.run_glm5_2_744b_a40b import (
        ScriptArgs,
        _convert_to_fp8,
        _execute_train,
        _prepare_download,
        _prepare_megatron_ckpt,
        _validate_glm_checkpoint,
    )
else:
    from scripts.run_glm5_2_744b_a40b import (
        ScriptArgs,
        _convert_to_fp8,
        _execute_train,
        _prepare_download,
        _prepare_megatron_ckpt,
        _validate_glm_checkpoint,
    )

# ROCm 注册迁移到 MI350 套件；在跑通之前保持 FIXME 禁用
register_rocm_ci(
    est_time=900,
    suite="stage-c-4-gpu-mi350",
```

```
    labels=["megatron", "model-scripts", "amd"],
    disabled="FIXME: re-enable once this case passes on the MI350 runners.",
)
```

评论区精华

Review 中最有价值的交锋集中在两处：

1. test_amd_r3_mtp.py 中 ep_size 从 4 改为 2，guapisolo 先问 “why change this?”，JessicaJiang 解释这是 Megatron 的并行约束：world_size=4、tp=2、pp=2、etp=1 时 expert grid size 为 8，4 不可整除，原用例配置本身就是 bug；guapisolo 随即认可 “oh that's a bug..” 和 “good fix !”。
2. miles/rollout/generate_hub/agentic_tool_call.py 的 use_v2 条件修改被 guapisolo 拒绝：“This shouldn't be modified like this. Tom has a follow-up PR fix this.”，表明该变更不应搭车进入 CI 主题 PR，应由专门的 agentic rollout 修复 PR 处理。

- test_amd_r3_mtp.py 中 ep_size 从 4 降到 2 的原因 (correctness): 确认原 AMD 用例并行配置错误，ep_size 必须降为 2 才能通过 Megatron 的 world_size 整除校验。
- agentic_tool_call.py 的 use_v2 条件修改不应搭车进入本 PR (design): 维护者在合并前由本人更新该文件定稿，将修改收敛为仅在多 leaf 时分配共享 rollout_id；后续由 #2536 等 PR 继续完善 agentic 两路径的一致性。

风险与影响

- 风险：
  1. agentic_tool_call.py 行为变更风险：该修改不属于本 PR 主题，且 review 中被质疑，虽然最终由维护者更新定稿，但仍需确认其与后续 #2536 的一致性，避免 v1/v2 rollout_id 语义回归。
  2. AMD 覆盖度不足：PR 合并时除 mori fp8 bridge 等少数用例外，绝大多数 AMD 用例仍处于 disabled 状态（blocked 于 sglang/aiter 上游多个 issue），CI 绿并不代表 AMD 训练已全面验证。
  3. 镜像未固定日期：提交切换到 “undated rocm720 image”，依赖刷新可能引入环境漂移；guapisolo 曾专门提交刷新镜像以包含仓库 pin 的 Transformers 版本，后续仍需关注版本一致性。
  4. 双份脚本漂移：scripts/amd/ 与 scripts/ 下的同名脚本需要同步维护，后续如果共享配方演进而 AMD 副本未跟进，可能导致两边行为悄悄分叉。
  5. 环境变量分发依赖：MILES_HARDWARE_PLATFORM 的取值与 Dockerfile.rocm 的注入强耦合，若环境变量误设，CUDA 测试也可能被错误分发到 scripts.amd，需要 CI 层保证隔离。 - 影响：对用户：NVIDIA 训练与推理行为完全不变，共享路径无 ROCm 分支。对系统：新增了 ROCm 专属 CI 通道与 MI350 runner 接入，AMD 平台的回归能力从无到有，为后续逐个启用被阻塞用例铺平道路。对团队：确立了“scripts/amd 配方 + MILES_HARDWARE_PLATFORM 分发 + extra_args/extra_env_vars 透传”的 AMD 支持模式，后续新 AMD 用例只需按该模式接入即可，同时 CI 基础设施向 MI350 套件迁移。 - 风险标记：ROCm 用例大量暂禁，agentic 行为变更争议，镜像未锁版本，双份脚本漂移，环境变量分发依赖

关联脉络

- PR #2536 Carry the rollout id on both agentic paths: guapisolo 在 review 中提到的“Tom 的 follow-up PR”，与本 PR 对 agentic_tool_call.py 的修改直接相关，继续修复 agentic v1/v2 的 rollout_id 不一致问题。
- PR #2265 [AMD] Enable Qwen3 FSDP hybrid-shard CI on ROCm: 本 PR 的关联 issue，为本 PR 提供了 ROCm 上 DeviceMesh._unflatten 缺失的能力回退背景，且本 PR 修改了 tests/e2e/fsdp/test_qwen3_4B_fsdp_hybrid_shard_r2s2.py。
- PR #2499 feat(ci): add weekly full-suite cadence: 同一 CI 基础设施演进线，修改了 tests/ci/ci_policy.py、tests/ci/run_suite.py、.github/workflows/pr-test.yml 等与本 PR 重叠的文件，共同塑造了 CI 套件与 runner 的组织方式。
- PR #2214 fix(ci): calibrate lora E2E estimates from nightly runs and halve GLM5 lora matrices: 同一测试体系下的 GLM5 E2E 用例调整，与本 PR 的 GLM-5.2 5-layer 用例共享 tests/e2e/megatron/model_scripts 目录，反映 AMD/NVIDIA 用例维护的协同。