

PR #2729 完整报告

radixark/miles

fix(ci): use local UCX transports for RDT test

合并时间: 2026-08-25 03:03

原文链接: <http://prhub.com.cn/radixark/miles/pull/2729>

执行摘要

- 一句话: 限制 RDT 测试使用本地 UCX 传输以修复 CI 失败
- 推荐动作: 值得精读, 尤其关注 `extra_env_vars` 的使用方式, 可作为类似测试环境限制的参考。

功能与动机

CI 测试 `stage-c-8-gpu-h100` 在 revision `1b756a756f` 上失败, 错误为

``uct_iface_open(rc_verbs/mlx5_bond_0:`

1) failed: Address not valid及NIXL_ERR_BACKEND。根本原因是 Novita H100 runner 暴露了不可用的`mlx5_bond_0:1``, 导致 UCX 自动发现中止, 无法选择同节点 CUDA IPC 路径。

实现拆解

1. 修改 `tests/e2e/megatron/test_qwen3_30B_A3B_rdt.py` 中的 `execute()` 函数, 在 `U.execute_train()` 调用中添加 `extra_env_vars` 参数, 设置 `UCX_TLS` 为 `sm,self,tcp,cuda_copy,cuda_ipc`。
2. 该环境变量通过 Ray runtime environment 传递给测试进程, 确保 UCX 只使用本地传输 (共享内存、自环、TCP、CUDA 复制、CUDA IPC), 避免 verbs 设备。
3. 保持生产 RDT 配置不变, 仅影响测试。
4. 验证: `py_compile` 和 `ruff check` 通过, 两 GPU Ray/NIXL H200 探针成功传输 `torch.arange(1024)` 并返回 `523776`。

关键文件:

- `tests/e2e/megatron/test_qwen3_30B_A3B_rdt.py` (模块 测试; 类别 `test`; 类型 `test-coverage`) : 为单节点 RDT E2E 测试添加 `UCX_TLS` 环境变量限制, 修复 CI 失败

关键符号: `execute`

关键源码片段

`tests/e2e/megatron/test_qwen3_30B_A3B_rdt.py`

为单节点 RDT E2E 测试添加 `UCX_TLS` 环境变量限制, 修复 CI 失败

```
# tests/e2e/megatron/test_qwen3_30B_A3B_rdt.py
```

```
# 在训练执行时，通过 extra_env_vars 显式限制 UCX 使用的传输层。
# 该单节点测试仅需 CUDA IPC，Novita H100 runner 上不可用的 mlx5 bond 会导致 UCX
后端创建失败。
U.execute_train(
    train_args=train_args,
    num_gpus_per_node=NUM_GPUS,
    megatron_model_type=MODEL_TYPE,
    train_script="train_async.py",
    # 限制传输为共享内存、自环、TCP、CUDA 复制和 CUDA IPC，避免 verbs 设备
    extra_env_vars={"UCX_TLS": "sm,self,tcp,cuda_copy,cuda_ipc"},
)
```

评论区精华

无实质 review 评论，仅有 Claude bot 的自动评论提示手动 review 设置，以及 yushengsu-thu 的批准。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。修改仅针对测试环境，通过 extra_env_vars 限制 UCX 传输，不影响生产 RDT 配置。但需要确保在 Novita H100 runner 上测试通过，且其他 RDT 测试或单节点场景不受影响。
- 影响：影响范围限于 RDT 相关 CI 测试，提升测试稳定性。对用户无直接影响，对团队 CI 流程有益。
- 风险标记：测试环境特定修复，未覆盖其他 RDT 测试

关联脉络

- PR #1313 RDT weight sync: GPU->GPU zero copy transfer through SGLang Ray actor backend: 引入 RDT 权重同步，本 PR 修改同一测试文件，属于相关功能线的后续修复