

PR #7827 完整报告

verl-project/verl

[megatron, ci] fix: configure Megatron integration in Ascend images and tests

合并时间: 2026-09-10 21:14

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7827>

执行摘要

- 一句话: Ascend 镜像与 NPU 单测默认加载 megatron_adaptor 并修正依赖 tag
- 推荐动作: 建议中低强度精读: 改动只有 11 行, 但位于「镜像构建 → patch → CI」的关键路径上, 维护 Ascend 镜像、NPU CI 与 Megatron 后端的同学花 5 分钟读完即可。最值得关注的设计决策有两处: 一是 VERL_USE_EXTERNAL_MODULES 是否应无条件写进镜像 ENV, 还是让下游启动脚本显式开启; 二是 Megatron-LM/MindSpeed 的 tag 引用是否应集中维护, 避免 Dockerfile 与安装脚本各写一份、日后再次漂移。

功能与动机

PR body 明确写出了动机: 「Load `megatron_adaptor` by default in the Ascend A2/A3 images and NPU unit tests through `VERL_USE_EXTERNAL_MODULES`. Correct the `Megatron-LM` ref to `core_v0.18.0` and simplify the `MindSpeed` clone while preserving its ref and full history.」结合 head 分支名 `fix_npu_unit_tests` 和标题前缀 `[megatron, ci] fix` 可以确认这是修复而非新增能力: 此前 Ascend 镜像与 NPU 单测没有默认加载 `megatron_adaptor`, `megatron-lm` 未被 patch; 同时 `Megatron-LM` 的 tag 引用写法有误。关联 Issue 列表与评论均为空, 动机只能从 PR body 与分支名还原。

实现拆解

1. 修正 Megatron-LM 依赖引用 (`docker/ascend/Dockerfile.ascend_9.1.0_a2`、`docker/ascend/Dockerfile.ascend_9.1.0_a3`): 克隆参数由 `--branch core_r0.18.0` 改为 `--branch core_v0.18.0`。NVIDIA/Megatron-LM 在 0.18.0 这一代 tag 带 v 前缀, 旧引用无法解析到目标版本, 是镜像构建与后续 patch 的直接失败点; 两个 Dockerfile 的改动完全对称, A2/A3 行为必须保持一致。
2. 简化 MindSpeed 拉取 (同上两个文件): 原先的 `git clone + cd MindSpeed && git checkout core_r0.18.0 && cd ..` 两段式被压缩为单条 `git clone --branch core_r0.18.0 https://gitcode.com/Ascend/MindSpeed.git`。这里刻意没有加 `--depth 1`, 即在 RUN 层内保留完整 git 历史, 对应 PR body 的 “preserving its ref and full history”——MindSpeed 的安装 / 打补丁流程依赖仓库元信息, 浅克隆会破坏它。
3. 镜像内默认加载 megatron_adaptor (同上两个文件): 在 `pip list` 之后、CMD `["/bin/bash"]` 之前新增 ENV `VERL_USE_EXTERNAL_MODULES=megatron_adaptor`。该变量在镜像内全局生效, verl 据此 import `megatron_adaptor` 去 patch `megatron-lm`, 因此用该镜像启动训练时不再需要手工 `export`; 代码注释写的 “Patch megatron-lm”

unconditionally”说明无条件开启是有意为之。

4. CI 环境对齐 (.github/workflows/npu_unit_tests.yml) : 在 job 级 env 中追加 VERL_USE_EXTERNAL_MODULES: "megatron_adaptor", 与镜像保持同一语义, 使 NPU 单测在不重建镜像、直接跑 workflow 的场景下也能得到正确的 megatron-lm 补丁, 避免「镜像能跑、CI 挂」的错配。
5. 测试与验证配套: 本 PR 未新增或修改任何测试、也未更新文档, 验证完全依赖既有 NPU 单测在 CI 中执行以及镜像重建, 属于配置类小改动的常见形态; 这也意味着回归信号依赖外部链路, PR 本身不提供自证。

关键文件:

- docker/ascend/Dockerfile.ascend_9.1.0_a2 (模块 昇腾镜像; 类别 infra; 类型 configuration; 符号 VERL_USE_EXTERNAL_MODULES, megatron_adaptor, core_v0.18.0) : A2 昇腾镜像的核心修复点: 修正 NVIDIA/Megatron-LM 的 tag 引用、简化 MindSpeed 拉取, 并新增 ENV VERL_USE_EXTERNAL_MODULES=megatron_adaptor 作为镜像默认行为。
- docker/ascend/Dockerfile.ascend_9.1.0_a3 (模块 昇腾镜像; 类别 infra; 类型 configuration; 符号 VERL_USE_EXTERNAL_MODULES, core_v0.18.0) : A3 昇腾镜像与 A2 同步的两处修复, 保证两种硬件形态下 Megatron 依赖引用与适配器加载行为一致。
- .github/workflows/npu_unit_tests.yml (模块 NPU 单测; 类别 infra; 类型 ci-config; 符号 VERL_USE_EXTERNAL_MODULES) : 让 NPU 单测 CI 与镜像使用同一套外部模块加载语义, 是「镜像改动能真正被 CI 验证」的关键配套。

关键符号: VERL_USE_EXTERNAL_MODULES (环境变量开关, 本 PR 无新增函数, 改动全部为环境 / 依赖配置), megatron_adaptor (外部模块入口, 由镜像 ENV 与 CI env 触发加载)

关键源码片段

`docker/ascend/Dockerfile.ascend_9.1.0_a2`

A2 昇腾镜像的核心修复点: 修正 NVIDIA/Megatron-LM 的 tag 引用、简化 MindSpeed 拉取, 并新增 ENV VERL_USE_EXTERNAL_MODULES=megatron_adaptor 作为镜像默认行为。

```
# 为便于阅读, 这里只保留 Megatron 相关的 clone 行, vllm / vllm-ascend 等行原样未动。
# 改动一: MindSpeed 由「先 clone 再 checkout core_r0.18.0」改成一次性 --branch 拉取,
# 并且刻意不加 --depth 1 —— 保留完整 git 历史, 因为 MindSpeed 的安装 / 打补丁流程
# 需要读取仓库元信息, 浅克隆会让这一步失败。
# 改动二: NVIDIA/Megatron-LM 在 0.18.0 这一代的 tag 带 v 前缀, 正确写法是 core_v0.18.0;
# 旧的 core_r0.18.0 在 NVIDIA 仓库并不存在, 会导致镜像构建失败或 patch 到错误版本。
RUN ARCH=$(uname -m) && \
    git clone --branch core_r0.18.0 https://gitcode.com/Ascend/MindSpeed.git && \
    git clone --depth 1 --branch core_v0.18.0 https://github.com/NVIDIA/Megatron-LM.git && \
    git clone --depth 1 --branch core_r0.18.0 https://gitcode.com/ascend/MegatronAdaptor.git && \
    git clone --depth 1 https://gitcode.com/ascend/TransformerEngineNPU.git && \
    git clone --depth 1 https://gitcode.com/ascend/MindSpeed-Ops.git

# VERL_USE_EXTERNAL_MODULES 作为镜像 ENV 全局生效: 容器内每个进程启动时,
```

```
# verl 都会据此 import megatron_adaptor 去 patch megatron-lm,  
# 因此 NPU 上跑 Megatron 后端不再需要手工 export;  
# 注释里的“unconditionally”说明即使只跑非 Megatron 后端也会加载。  
ENV VERL_USE_EXTERNAL_MODULES=megatron_adaptor
```

```
# Setting Default Commands  
CMD ["/bin/bash"]
```

[.github/workflows/npu_unit_tests.yml](#)

让 NPU 单测 CI 与镜像使用同一套外部模块加载语义，是「镜像改动能真正被 CI 验证」的关键配套。

```
# .github/workflows/npu_unit_tests.yml —— 在 job 级 env 注入与镜像一致的变量  
jobs:  
  npu_unit_tests:  
    env:  
      HF_ENDPOINT: "https://hf-mirror.com"  
      HF_HUB_ENABLE_HF_TRANSFER: "0" # This is more stable  
      # 本次新增：CI runner 上默认加载 megatron_adaptor，使 megatron-lm 被打上 NPU 侧补丁，  
      # 与 Dockerfile 中的 ENV VERL_USE_EXTERNAL_MODULES 保持同一语义，  
      # 避免出现「镜像里能跑、CI 里跑挂」的错配。  
      VERL_USE_EXTERNAL_MODULES: "megatron_adaptor" # import megatron_adaptor to patch  
      megatron-lm  
    steps:  
      - name: Check npu and CANN info
```

评论区精华

PR 没有任何 inline review 评论与讨论线程（comments_count = 0、review_comments_count = 0）。唯一的审核记录是 wucong25 的 APPROVED，body 为空，属于静默批准：既没有对 tag 改名、ENV 全局生效的争议，也没有留下待办意见。因此本 PR 的“讨论价值”不在于交锋，而在于几个未被讨论的隐含决策——尤其是把 **VERL_USE_EXTERNAL_MODULES** 写死为镜像 ENV、对非 Megatron 后端用户也生效这一点。

- 无条件在镜像内开启 VERL_USE_EXTERNAL_MODULES 的影响范围 (design): 未在 review 中讨论，实际以「镜像全局生效」的形态合并；属于有意为之但缺少显式确认。
- Megatron-LM tag 由 core_r0.18.0 更正为 core_v0.18.0 (correctness): 已修正，A2/A3 两个 Dockerfile 保持一致；未验证其他昇腾安装脚本是否也存在同类旧写法。

风险与影响

- 风险：1) ENV 全局无条件生效：VERL_USE_EXTERNAL_MODULES=megatron_adaptor 写死在镜像 ENV，所有基于该镜像的进程都会 import 并 patch megatron-lm，对只跑 FSDP/VeOmni 的昇腾用户是隐式行为变化；若用户还想叠加其他 external modules，需覆盖整个变量而非追加，容易踩坑（多值语法未在本 diff 中体现）。2) 镜像体积与构建耗时：MindSpeed 不再浅克隆、保留完整历史，镜像层变大、clone 变慢，是为满足安装流程付出的代价。3) 引用一致性：tag 修正只覆盖两个 Dockerfile，若

scripts/install_vllm_mcore_npu.sh 等其他昇腾安装脚本仍沿用旧写法，可能出现 Megatron 版本漂移，需要后续确认。4) 验证强度不足：无新增测试、无 review 讨论，正确性只靠 NPU 单测与镜像重建间接验证，ENV 名称拼写或解析语义偏差只会在 NPU 用例上暴露。5) 影响面：仅限 Ascend A2/A3 镜像与 NPU 单测工作流，不触及 GPU 侧默认行为与主训练路径。

- 影响：用户侧：使用 Ascend A2/A3 镜像的昇腾用户会默认获得打过 patch 的 megatron-lm，Megatron 后端开箱可用；同时非 Megatron 后端的用户会在无感知的情况下多加载一个外部模块。系统侧：直接影响 NPU 单测稳定性与镜像构建可重复性，是昇腾 Megatron 链路的必要前置条件。团队侧：后续涉及 Ascend + Megatron 的 PR 需要记住该镜像 ENV 的隐式前置，排障时也应优先确认 VERL_USE_EXTERNAL_MODULE S 是否被用户或启动脚本覆盖。
- 风险标记：ENV 全局无条件生效，无自动化验证，无 review 讨论，镜像体积与构建耗时增加，其他昇腾安装脚本可能引用旧 tag

关联脉络

- PR #7604 [megatron, ci] chore: upgrade Ascend stack to Megatron 0.18.0: 该 PR 把昇腾栈升级到 Megatron 0.18.0 并改动同一批 Ascend Dockerfile 与安装脚本，本 PR 是这次升级的收尾修复：修正遗留的 tag 写法与适配器加载入口。
- PR #7799 [ci] fix: nightlyCI_grpo_qwen3_5_2b_fsdp2_vllm_ascend: 同为昇腾 + Megatron 的 CI 链路收敛，思路一致地减少手工安装步骤、改由镜像 / 环境变量统一承载。
- PR #7779 [vllm, hardware] fix: patch vllm-ascend for GLM-5.2 RL training on Ascend NPUs: 同样是在 Ascend 侧通过补丁 / 环境手段打通 Megatron 与推理后端，属于同一条昇腾训练可达性主线。
- PR #7788 [recipe] feat: add GLM-5.2 GRPO training example on Ascend NPUs: 昇腾 + Megatron 的示例落地依赖本 PR 修复的镜像与 patch 前置条件，是同一演进方向的下游消费方。