

# PR #7042 完整报告

verl-project/verl

[ci] chore: remove vllm\_ascend patch and fix ci

合并时间: 2026-07-15 23:09

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7042>

## 执行摘要

- 一句话: 移除 vLLM Ascend 旧补丁并修复 CI
- 推荐动作: 建议团队成员了解此变更, 特别是涉及 Ascend 平台开发和测试的成员。此 PR 展示了如何跟踪上游依赖变化并及时清理过期补丁, 是良好的工程实践。审查时应重点确认所有依赖旧补丁的代码路径已完全移除, 并运行 Ascend CI 测试以验证兼容性。

## 功能与动机

随着 vLLM 和 vllm\_ascend 的版本演进, 原有针对 v0.11/v0.13 的 MoE 通信方法包装器等补丁已不再需要, 移除这些过时代码可减少维护负担并修复因依赖过期导致的 CI 失败。

## 实现拆解

1. 删除旧补丁函数: 在 npu\_vllm\_patch.py 中删除 vllm\_ascend\_v011\_select\_moe\_comm\_method\_wrapper、vllm\_ascend\_v011\_matmul\_and\_reduce\_wrapper、vllm\_ascend\_v013\_select\_moe\_comm\_method\_wrapper、vllm\_ascend\_v013\_matmul\_and\_reduce\_wrapper、check\_vllm\_ascend\_before\_server\_launch 及辅助检测函数, 仅保留面向 v0.13+ 的权重加载包装器和旋转嵌入补丁。
2. 清理调用点: 在 vllm\_async\_server.py 的 \_apply\_quantization 方法中移除对 check\_vllm\_ascend\_before\_server\_launch 的条件导入和调用。
3. 更新导出口: 在 verl/utils/vllm/\_\_init\_\_.py 中删除 check\_vllm\_ascend\_before\_server\_launch 的导入和 \_\_all\_\_ 条目。
4. 同步 Docker 构建: 在 Dockerfile.ascend.sglang\_8.5.0\_a2、Dockerfile.ascend.sglang\_8.5.0\_a3、Dockerfile.ascend\_9.0.0\_a2、Dockerfile.ascend\_9.0.0\_a3 中移除 Megatron-Bridge 仓库克隆步骤和对应的 PYTHONPATH 环境变量设置。

关键文件:

- verl/utils/vllm/npu\_vllm\_patch.py (模块 NPU 补丁; 类别 source; 类型 dependency-wiring; 符号 vllm\_ascend\_v011\_select\_moe\_comm\_method\_wrapper, vllm\_ascend\_v011\_matmul\_and\_reduce\_wrapper, check\_vllm\_ascend\_before\_server\_launch, \_is\_ascend\_soc\_version\_A2\_v011\_local) : 核心变更文件: 删除了 165 行旧补丁代码, 引入了新的 v0.13+ 权重加载包装器和旋转嵌入

补丁，是本次清理的主体。

- verl/workers/rollout/vllm\_rollout/vllm\_async\_server.py (模块 Rollout; 类别 source; 类型 dependency-wiring) : 移除了对已删除的 check\_vllm\_ascend\_before\_server\_launch 函数的条件导入和调用，这是保证清理后不会在 NPU 环境下报错的关键配套修改。
- verl/utils/vllm/\_\_init\_\_.py (模块 VLLM 工具; 类别 source; 类型 dependency-wiring) : 移除了对 check\_vllm\_ascend\_before\_server\_launch 的公共导出，保持 API 清洁。
- docker/ascend/Dockerfile.ascend.sglang\_8.5.0\_a2 (模块 Docker 构建; 类别 infra; 类型 infrastructure) : 移除 Megatron-Bridge 仓库克隆和 PYTHONPATH 设置，反映依赖变更。
- docker/ascend/Dockerfile.ascend.sglang\_8.5.0\_a3 (模块 Docker 构建; 类别 infra; 类型 infrastructure) : 与 a2 版本同步移除 Megatron-Bridge 相关构建步骤。
- docker/ascend/Dockerfile.ascend\_9.0.0\_a2 (模块 Docker 构建; 类别 infra; 类型 infrastructure) : 同步移除 Megatron-Bridge 并替换为 transformers 克隆，更新构建依赖。
- docker/ascend/Dockerfile.ascend\_9.0.0\_a3 (模块 Docker 构建; 类别 infra; 类型 infrastructure) : 完全移除 Megatron-Bridge 相关行，与 a2 版本对齐。

关键符号: vllm\_ascend\_v011\_select\_moe\_comm\_method\_wrapper,  
vllm\_ascend\_v011\_matmul\_and\_reduce\_wrapper,  
check\_vllm\_ascend\_before\_server\_launch, \_is\_ascend\_soc\_version\_A2\_v011\_local,  
\_is\_ascend\_soc\_version\_A2\_v013\_local, vllm\_ascend\_v013\_select\_moe\_comm\_method\_w  
rapper, vllm\_ascend\_v013\_matmul\_and\_reduce\_wrapper,  
vllm\_v013\_weight\_loader\_method\_wrapper, patch\_vllm013\_rotary\_emb

## 关键源码片段

### verl/utils/vllm/npu\_vllm\_patch.py

核心变更文件: 删除了 165 行旧补丁代码，引入了新的 v0.13+ 权重加载包装器和旋转嵌入补丁，是本次清理的主体。

# 变更后的 npu\_vllm\_patch.py 保留内容 (移除旧补丁后)

```
from functools import wraps
```

```
from verl.utils.device import is_torch_npu_available
```

```
def vllm_v013_weight_loader_method_wrapper(fn):
```

```
    '''针对 v0.13+ 的 FusedMoE 权重加载包装器，  
    在加载前对权重矩阵进行转置以匹配 Ascend NPU 内存布局。'''
```

```
    @wraps(fn)
```

```
    def wrapper(self, param, loaded_weight, weight_name, shard_id, expert_id, return_success=  
False):
```

```
        # 判断是否需要对权重进行转置: expert 权重形状与 hidden_size 相关
```

```
        if (shard_id in ('w1', 'w3') and param.shape[1] == self.hidden_size) or (
```

```
            shard_id == 'w2' and param.shape[2] == self.hidden_size
```

```

):
    param.data = param.data.transpose(1, 2)
    return fn(self, param, loaded_weight, weight_name, shard_id, expert_id, return_success)
return wrapper

```

```

def patch_vllm013_rotary_emb():
    '''对 v0.13+ 的打补丁：禁用 ApplyRotaryEmb 中的 flash_attn,
    因为 Ascend NPU 目前不支持该模式。'''
    from vllm.model_executor.layers.rotary_embedding.common import ApplyRotaryEmb

    def vllm013_npu_rotary_embedding_init_impl(
        self,
        enforce_enable: bool = False,
        is_neox_style: bool = True,
        enable_fp32_compute: bool = False,
    ) -> None:
        super(ApplyRotaryEmb, self).__init__()
        self.is_neox_style = is_neox_style
        self.enable_fp32_compute = enable_fp32_compute
        self.apply_rotary_emb_flash_attn = None # 禁用 flash_attn

    ApplyRotaryEmb.__init__ = vllm013_npu_rotary_embedding_init_impl

if is_torch_npu_available(check_device=False):
    import vllm
    from packaging import version

    _VLLM_VERSION = version.parse(vllm.__version__)
    # 对 v0.13 ~ v0.14 和 v0.18+ 分别打补丁
    if (_VLLM_VERSION >= version.parse('0.13.0') and _VLLM_VERSION <= version.parse('0.14.
0')) or \
        (_VLLM_VERSION >= version.parse('0.18.0')):
        from vllm.model_executor.layers.fused_moe import FusedMoE
        patch_vllm013_rotary_emb()
        FusedMoE.weight_loader = vllm_v013_weight_loader_method_wrapper(FusedMoE.weight_
loader)

```

## 评论区精华

Gemini Code Assist 机器人指出，在删除 `check_vllm_ascend_before_server_launch` 时，`vllm_async_server.py` 中仍然存在导入和调用，将导致 `ImportError`，建议同步清理。该评论被采纳，在后续提交中已移除对应代码。

- 删除 `check_vllm_ascend_before_server_launch` 导致 `ImportError (correctness)`: 作者接受了反馈，在后续提交（fix import、fix ci）中移除了 `vllm_async_server.py` 中的相关代码，确保了无遗漏。

## 风险与影响

- 风险：
  - Ascend 兼容性风险：如果某些 Ascend 环境仍依赖旧版 `vllm_ascend` ( $<0.13$ )，移除补丁可能导致 MoE 通信选择错误或显存优化失效。但 PR 上下文暗示团队已验证新版本不再需要。
  - Docker 一致性风险：移除 Megatron-Bridge 可能影响依赖该桥接库的 Megatron 训练流程；但 Dockerfile 中保留了 MindSpeed 和 Megatron-LM，推测已迁移到其他方案。
  - 遗漏引用风险：尽管已清理 `vllm_async_server.py`，仍需确认其他潜在引用（如 `npu_vllm_patch.py` 是否被其他地方直接导入）。
- 影响：
  - 用户影响：不再需要手动应用 Ascend 特定 MoE 补丁，代码更整洁；但需确保使用兼容的 vLLM 版本 ( $\geq 0.13$ )。
  - 系统影响：Docker 镜像体积略有减小（移除一次 git clone），构建速度可能提升。
  - 团队影响：减少维护点，但需要确保所有 Ascend 训练场景（如单卡多进程 MC2）已在新版本中得到支持。
  - 影响程度：中等，主要影响 Ascend NPU 用户，其他不受影响。
  - 风险标记：Ascend 兼容性，依赖清理遗漏，Docker 一致性

## 关联脉络

- PR #7039 [env] chore: add megatron-bridge in sglang docker env: 本 PR 移除了 #7039 刚刚添加的 Megatron-Bridge 相关设置，形成了一次反转操作，可能反映了依赖策略的调整。
- PR #7035 [ci] test: update ascend ci docker tag: 都与 Ascend CI Docker 构建相关，体现了 Ascend 基础设施的持续演进。