PR #7190 完整报告

verl-project/verl

[vllm] refactor: drop support for vLLM older than 0.18.0

合并时间：2026-07-30 10:19

原文链接：http://prhub.com.cn/verl-project/verl/pull/7190

执行摘要

- 一句话：提升 vLLM 最低支持版本至 0.18.0，移除旧版兼容代码
- 推荐动作：建议精读 verl/utils/vllm/vllm_fp8_utils.py 的改动，了解 FP8 权重处理的版本依赖是如何被剥离的，同时关注 NPU 补丁路径的简化是否覆盖所有场景。

功能与动机

PR body 指出：vLLM rollout 路径携带了从 0.7.0 开始的兼容分支，而代码本身已经向前兼容到 0.22.0，且声明的安装范围（vllm>=0.8.5,<=0.12.0）和实际处理的版本范围不一致。通过提升最低版本到 0.18.0，可以清理大量无用的兼容性代码，简化维护。

实现拆解

1. 更新第三方包版本检查逻辑：在 verl/third_party/vllm/__init__.py 中，将最低版本判断从 0.7.0 提升至 0.18.0，并移除针对 0.5.4 和 0.6.3 的明确拒绝分支，同时统一了 sleep level 的默认值。
2. 清理 FP8 工具中的版本分支：在 verl/utils/vllm/vllm_fp8_utils.py 中，删除 process_weights_after_loading_for_vllm10、process_weights_after_loading_for_vllm11、process_weights_after_loading_moe_for_vllm10、process_weights_after_loading_moe_for_vllm11 四个函数以及 _create_param_from_subclass_attributes 辅助函数，简化 quant_weights 中 scale 后缀的版本判断逻辑，直接统一使用 _scale_inv。
3. 简化启动服务器中的版本判断：在 verl/workers/rollout/vllm_rollout/vllm_async_server.py 中，移除 _VLLM_VERSION 相关的版本条件导入和 _RESET_PREFIX_CACHE_KWARGS，直接导入 FlexibleArgumentParser 和 get_encoding，并简化 cudagraph_capture_sizes 和 profiler 参数的处理逻辑。
4. 移除 sleep level 版本检测：在 verl/workers/rollout/vllm_rollout/vllm_rollout.py 中，删除 _check_vllm_version_for_sleep_level 函数，直接使用 VLLM_SLEEP_LEVEL 常量，不再根据 vLLM 版本动态调整。
5. 更新依赖声明和安装脚本：在 setup.py 中将 vLLM 依赖修改为 vllm>=0.18.0，移除上限；在 scripts/install_vllm_sglang_mcore.sh 中更新安装的 vLLM 版本为 0.18.0 以上；在 docs/start/install.rst 中同步更新文档示例。

6. 简化 NPU 补丁路径：在 verl/utils/vllm/npu_vllm_patch.py 中，移除对 0.13-0.14 分支的特殊处理，统一执行 rotary embedding 和 fused MoE 的补丁。

关键文件：

- verl/utils/vllm/vllm_fp8_utils.py（模块 FP8 工具；类别 source；类型 dependency-wiring；符号 process_weights_after_loading_for_vllm10, _create_param_from_subclass_attributes, process_weights_after_loading_for_vllm11, process_weights_after_loading_moe_for_vllm10）：核心文件，删除了 4 个版本兼容函数和大量分支逻辑，修改了 quant_weights 中的 scale 后缀判断，是最主要的清理对象。
- verl/workers/rollout/vllm_rollout/vllm_async_server.py（模块 VLLM 服务器；类别 source；类型 core-logic；符号 _validate_launch_requirements）：大幅简化了版本条件导入和启动参数处理，移除 _cuda_graph_sizes 版本分支。
- verl/workers/rollout/vllm_rollout/vllm_rollout.py（模块 VLLM 适配器；类别 source；类型 core-logic；符号 _check_vllm_version_for_sleep_level）：删除 _check_vllm_version_for_sleep_level 函数，简化 sleep level 判断逻辑。
- verl/utils/vllm/npu_vllm_patch.py（模块 NPU 补丁；类别 source；类型 dependency-wiring）：移除版本条件分支，统一 NPU 补丁路径。
- verl/third_party/vllm/__init__.py（模块 三方管理；类别 source；类型 core-logic）：修改版本检查逻辑，提升最低版本并移除无效分支。
- setup.py（模块 安装配置；类别 source；类型 configuration）：更新 vLLM 依赖声明，移除版本上限。
- scripts/install_vllm_sglang_mcore.sh（模块 安装脚本；类别 other；类型 infra）：同步更新安装脚本中的 vLLM 版本。
- docs/start/install.rst（模块 文档；类别 docs；类型 documentation）：更新文档中的版本引用。

关键符号：process_weights_after_loading_for_vllm10, process_weights_after_loading_for_vllm11, process_weights_after_loading_moe_for_vllm10, process_weights_after_loading_moe_for_vllm11, _create_param_from_subclass_attributes, _validate_launch_requirements, _check_vllm_version_for_sleep_level, quant_weights, apply_npu_vllm_patches, get_version

关键源码片段

verl/utils/vllm/vllm_fp8_utils.py

核心文件，删除了 4 个版本兼容函数和大量分支逻辑，修改了 quant_weights 中的 scale 后缀判断，是最主要的清理对象。

```
# verl/utils/vllm/vllm_fp8_utils.py
# 简化后的 quant_weights 函数：移除了版本判断，统一使用 _scale_inv

def quant_weights(weights, model, quant_config, dtype=torch.bfloat16):
    """Quantize weights to FP8 format using a memory-efficient generator."""
```

```
    if is_deepseek_v4_model(model):
        yield from iter_deepseek_v4_weights(weights)
        return

    fp8_state.seen_params.clear()
    fp8_state.fp8_param_names.clear()
    is_mxfp8_npu = is_mxfp8_vllm_ascend(quant_config)
    if is_mxfp8_npu:
        import torch_npu

    for k, v in weights:
        if not is_fp8_weight(k, model):
            yield (k, v)
            continue

        # 执行量化
        if is_mxfp8_npu:
            param_lp, param_scale = torch_npu.npu_dynamic_mx_quant(
                v.to(dtype), axis=-1, dst_type=torch_npu.float8_e4m3fn,
            )
            param_scale = param_scale.flatten(-2, -1)
        else:
            param_lp, param_scale = scaled_fp8_blockwise(
                v.to(dtype), weight_block_size=quant_config.weight_block_size,
            )
        param_scale = param_scale.squeeze(-1)

        yield (k, param_lp)

        # 之前需要根据版本决定使用 _scale 还是 _scale_inv，现在统一用 _scale_inv
        if is_mxfp8_npu:
            yield (k + "_scale", param_scale)
        else:
            yield (k + "_scale_inv", param_scale)

        del v, param_lp, param_scale
```

verl/workers/rollout/vllm_rollout/vllm_async_server.py

大幅简化了版本条件导入和启动参数处理，移除 _cuda_graph_sizes 版本分支。

```
# verl/workers/rollout/vllm_rollout/vllm_async_server.py
# 简化后的导入和版本判断：直接导入，无需条件分支

from vllm.entrypoints.openai.parser.harmony_utils import get_encoding
from vllm.utils.argparse_utils import FlexibleArgumentParser

# 之前根据版本有条件导入，现在直接导入，因为最低版本要求 0.18.0

if os.getenv("VERL_USE_GPT_OSS", "0") == "1":
```

```
    get_encoding()

# 启动参数中：移除 cudagraph_capture_sizes 的版本分支
if self.config.cudagraph_capture_sizes:
    compilation_config["cudagraph_capture_sizes"] = self.config.cudagraph_capture_sizes

# 之前需要判断版本，现在直接添加
```

评论区精华

无实质性讨论，代码被直接批准（reviewer wuxibin89 已批准）。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险是向下不兼容：使用 vLLM 0.18.0 以下版本的用户必须强制升级。同时，大量兼容代码的删除可能隐藏某些边缘情况下的回归，尤其是在 FP8 权重加载、NPU 环境下的 rotary embedding 补丁路径。另外，未添加对应的回归测试，仅依赖已有测试覆盖。
- 影响：影响所有使用 vLLM rollout 的用户，必须升级 vLLM 到 0.18.0+；代码库维护者受益于清理，但需注意可能遗漏的旧版本兼容场景。影响程度高，但属合理的技术债务清理。
- 风险标记：向下不兼容，核心路径变更，缺少测试覆盖

关联脉络

- PR #7179 [vllm] refactor: clean up weight sync: 同为 vLLM 相关重构，清理权重同步逻辑，与本 PR 的清理目标一致。
- PR #7101 [docker] feat: upgrade vllm and megatron version, add packages to support DeepSeek-V4: 涉及 vLLM 版本升级，与本 PR 的版本底线上调相关联。