PR #6789 完整报告

verl-project/verl

[data] fix: offload process_vision_info to a thread executor to unblock the agent-loop event loop

合并时间：2026-06-17 22:30

原文链接：http://prhub.com.cn/verl-project/verl/pull/6789

执行摘要

- 一句话：将 vision info 处理卸到线程池避免阻塞事件循环
- 推荐动作：该 PR 思路正确，改动清晰。建议后续补充对 process_multi_modal_info 的同类修复以彻底解决问题。PR 值得快速合入。

功能与动机

在 AgentLoop 中，多个 trajectory coroutine 共享同一个事件循环，但 process_vision_info 内部同步调用了 qwen_vl_utils.process_vision_info（PNG 解码 + smart_resize），阻塞了事件循环，导致其它协程的 socket I/O 被饿死（表现为连接超时 / 零窗口暂停）。需要将此 CPU 密集型任务卸到线程池。

实现拆解

1. 在 verl/utils/dataset/rl_dataset.py 的 import 部分添加 import asyncio，引入异步支持。
2. 修改 RLHFDataset.process_vision_info 方法：将原来直接同步调用 process_vision_info 改为通过 asyncio.get_running_loop().run_in_executor(None, lambda: ...) 在默认线程池中执行。
3. 由于 PIL/numpy 在解码 / 缩放时会释放 GIL，卸到线程池后既能让事件循环保持响应，又能让负载在线程间并行。
4. 该方法只会在 AgentLoop（始终有运行中事件循环）中被 await，因此 asyncio.get_running_loop() 安全可用。

关键文件：

- verl/utils/dataset/rl_dataset.py（模块 数据；类别 source；类型 core-logic；符号 process_vision_info）：唯一修改文件，核心变更位置。将同步 CPU 密集型调用卸到线程池，防止阻塞事件循环。

关键符号：process_vision_info

关键源码片段

verl/utils/dataset/rl_dataset.py

唯一修改文件，核心变更位置。将同步 CPU 密集型调用卸到线程池，防止阻塞事件循环。

```
# verl/utils/dataset/rl_dataset.py
```

```
import asyncio # 新增导入
import copy
import logging
from typing import Any, Optional

class RLHFDataset:

    @classmethod
    async def process_vision_info(
        cls,
        messages: list[dict],
        image_patch_size,
        config: DictConfig,
    ) -> tuple[list[Image.Image], list[tuple[torch.Tensor, dict]]]:
        """从消息中提取图片和视频。此方法由 AgentLoop 调用。

        原先直接调用同步的 qwen_vl_utils.process_vision_info (PNG 解码 +
        smart_resize, CPU 密集), 会阻塞共享的事件循环。
        现将其卸到默认线程池，PIL/numpy 在解码时会释放 GIL,
        从而让事件循环保持响应，并且负载可以并行。
        """
        from qwen_vl_utils import process_vision_info

        # 获取当前运行的事件循环
        loop = asyncio.get_running_loop()
        # 在线程池中执行 CPU 密集操作
        images, videos = await loop.run_in_executor(
            None, # 使用默认线程池
            lambda: process_vision_info(
                messages,
                image_patch_size=image_patch_size,
                return_video_metadata=True,
            ),
        )
        return images, videos
```

评论区精华

Review 中 gemini-code-assist[bot] 指出类似的阻塞问题同样存在于 process_multi_modal_info 方法中——该方法是 agent loop 实际调用的入口，仍然同步执行。建议也将其卸到线程池以彻底解决阻塞问题。当前 PR 仅修复了 process_vision_info，process_multi_modal_info 的阻塞问题仍然存在。

- process_multi_modal_info 同样存在阻塞 (correctness): 当前 PR 仅修复了 process_vision_info，需要后续补充对 process_multi_modal_info 的同样处理。

风险与影响

- 风险:

1. 回归风险低：仅修改一个类方法的实现细节，外部接口（async 签名、参数、返回值）保持不变。
2. 若默认线程池被占满，卸到线程池可能增加延迟；但当前场景下 CPU 密集调用量不大，风险可控。
3. 未处理 process_multi_modal_info 的类似问题，可能导致不完全修复。 - 影响：影响范围窄，仅影响多模态 agent loop 场景下调用 RLHFDataset.process_vision_info 的路径。能够缓解事件循环阻塞问题，提升并发吞吐。对单协程场景无影响。 - 风险标记：部分修复，低风险

关联脉络

- 暂无明显关联 PR