PR #2050 完整报告

THUDM/slime

Set RAY_USE_UVLOOP=0 for Ray actors

合并时间：2026-06-11 10:36

原文链接：http://prhub.com.cn/THUDM/slime/pull/2050

执行摘要

- 一句话：为所有 Ray Actor 设置 RAY_USE_UVLOOP=0
- 推荐动作：值得快速合并，改动清晰、风险低、收益明确。建议作为生产环境的基础稳定性补丁。后续可关注 Ray 社区关于 uvloop 的修复进展，适时移除该 workaround。

功能与动机

Ray 的 uvloop 集成在异步 Actor 环境中会引发间歇性问题（代码注释明确说明 'Ray's uvloop integration has caused intermittent async actor issues'），导致 actor 意外卡死或响应超时。通过设置 RAY_USE_UVLOOP=0 显式禁用 uvloop，可在不修改核心逻辑的前提下提升系统稳定性。

实现拆解

1. 核心函数新增：在 slime/ray/utils.py 中新增常量字典 RAY_DEFAULT_ENV_VARS = {'RAY_USE_UVLOOP': '0'} 及函数 add_default_ray_env_vars(env_vars=None)，该函数返回 RAY_DEFAULT_ENV_VARS 与传入环境变量的合并结果，确保默认值优先级最高。
2. 调用点适配：在 6 个文件中导入并调用 add_default_ray_env_vars，统一注入到 runtime_env['env_vars'] 中： - slime/ray/rollout.py：在 RolloutManager.__init__ 和 RolloutEngineGroup.start_engines 中，将原有的 env_vars 包裹一层调用。 - slime/backends/sglang_utils/external.py：在外部 rollout 引擎启动时，为 RolloutRayActor 添加 runtime_env。 - slime/ray/actor_group.py：在 RayTrainGroup._allocate_gpus_for_actor 中，为 Megatron 训练 Actor 注入。 - slime/ray/placement_group.py：在 RolloutManager 和 Information Actor 创建时注入。 - slime/utils/http_utils.py：在 _HttpPosterActor 创建时注入。 - slime/utils/external_utils/command_utils.py：在命令执行相关 Actor 中注入。
3. 无额外配套改动：本次变更不涉及测试、配置或部署文件的修改，仅通过源码级别的环境变量注入解决问题。

关键文件：

- slime/ray/utils.py（模块 核心工具；类别 source；类型 core-logic；符号 add_default_ray_env_vars）：核心变更文件：新增 RAY_DEFAULT_ENV_VARS 常量和 add_default_ray_env_vars 函数，是全局注入的源头。

- slime/ray/rollout.py（模块 Rollout 管理；类别 source；类型 dependency-wiring）：关键调用方：Rollout 引擎和 Lock 的创建路径使用该函数，覆盖了主要的 rollout 工作流。
- slime/backends/sglang_utils/external.py（模块 外部引擎；类别 source；类型 dependency-wiring）：外部引擎启动路径：确保外部 SGLang 引擎也应用默认环境变量。
- slime/ray/actor_group.py（模块 Actor 管理；类别 source；类型 dependency-wiring）：训练 Actor 创建路径：为 Megatron 训练 Actor 注入默认环境变量。
- slime/ray/placement_group.py（模块 Placement 管理；类别 source；类型 dependency-wiring）：Placement Group 创建路径：为 RolloutManager 和 Information Actor 注入默认环境变量。
- slime/utils/http_utils.py（模块 HTTP 工具；类别 source；类型 dependency-wiring）：HTTP 工具路径：为 _HttpPosterActor 注入默认环境变量。
- slime/utils/external_utils/command_utils.py（模块 命令工具；类别 source；类型 dependency-wiring）：命令执行路径：为外部命令 Actor 注入默认环境变量。

关键符号：add_default_ray_env_vars

关键源码片段

slime/ray/utils.py

核心变更文件：新增 RAY_DEFAULT_ENV_VARS 常量和 add_default_ray_env_vars 函数，是全局注入的源头。

```
# Adapted from OpenRLHF
import os
import ray
import torch
from slime.ray.ray_actor import RayActor

# 列举 Ray 不会自动设置的可见设备环境变量（用于 GPU 设备绑定）
NOSET_VISIBLE_DEVICES_ENV_VARS_LIST = [
    "RAY_EXPERIMENTAL_NOSET_CUDA_VISIBLE_DEVICES",
    "RAY_EXPERIMENTAL_NOSET_ROCR_VISIBLE_DEVICES",
    # ... 其他硬件相关变量
]

# 所有 Ray Actor 应默认使用的环境变量
# Ray 的 uvloop 集成曾导致间歇性异步 Actor 问题，故显式禁用
RAY_DEFAULT_ENV_VARS = {
    "RAY_USE_UVLOOP": "0",
}


def add_default_ray_env_vars(env_vars: dict[str, str] | None = None) -> dict[str, str]:
    """将默认环境变量合并到给定的 env_vars 中（默认变量优先级更高）。"""
    return RAY_DEFAULT_ENV_VARS | (env_vars or {})
```

评论区精华

该 PR 没有 review 评论或讨论，变更由作者直接合并。核心决策（新增统一注入点而非逐个硬编码）体现了对可维护性的考量。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。RAY_USE_UVLOOP=0 是 Ray 的官方环境变量，禁用 uvloop 仅影响异步性能（略微增加上下文切换开销），不会影响功能正确性。回滚只需删除相关行即可。但需注意：如果未来 Ray 版本修复了 uvloop 问题，需要手动移除该环境变量。
- 影响：影响范围覆盖系统中所有 Ray Actor 的创建路径，包括 rollout engine、train actor、lock、http poster 等，但对用户无可见功能变化。性能方面可能带来微小的异步 I/O 延迟增加，但在 GPU 密集型训练 / 推理场景中可忽略。稳定性方面显著降低 Actor 因 uvloop 卡死的概率。
- 风险标记：无回归风险，无测试覆盖

关联脉络

- PR #2046 Revert "[docker] always re-register mooncake addr during offloading": 均为稳定性修复型 PR，但涉及不同模块（uvloop vs mooncake）。