PR #6664 完整报告

verl-project/verl

[rollout, sglang] feat: support sglang ROCm backend via aiter defaults and ray init env

合并时间：2026-06-16 15:32

原文链接：http://prhub.com.cn/verl-project/verl/pull/6664

执行摘要

- 一句话：SGLang ROCm 后端开箱即用支持
- 推荐动作：建议精读讨论中关于 ROCm 设备可见性优先级的部分，对理解 Ray 在 ROCm 上的行为差异非常有价值。设计上值得关注的是将平台特定环境变量迁移到 platform 层的方法。

功能与动机

SGLang rollout 后端此前在 AMD ROCm GPU 上无法直接运行，因为 attention 后端默认选用了 CUDA 专属的 flashinfer/fa3，且 Ray 在 ROCm 上对 num_gpus=0 actor 会清除 HIP_VISIBLE_DEVICES，导致 SGLang server actor 无可用 GPU。PR 目标是让 SGLang 在 ROCm 上无需用户额外配置即可工作，同时保持用户可覆盖性。

实现拆解

1. 默认 attention 后端适配 (async_sglang_server.py)：在 launch_server 方法中，当用户未指定 attention_backend 时，通过 torch.version.hip is not None 判断 ROCm 平台并默认设为 "aiter"，而非 CUDA 下的 flashinfer/fa3。
2. 环境变量注入迁移到 platform 层 (async_sglang_server.py)：SGLang server actor 的 runtime_env 现在通过 get_platform().ray_noset_envvars() + get_platform().rollout_env_vars() 获取平台特定的环境变量（如 SGLANG_USE_AITER=1），替代了原来的硬编码和 os.environ.setdefault，与 vLLM/trtllm rollout server 保持一致。
3. 修复 ROCm 下 Ray 的加速器可见性 (constants_ppo.py)：新增 _rocm_ray_env 字典，在 ROCm 平台下注入 RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO=0 到 PPO_RAY_RUNTIME_ENV，阻止 Ray 对 num_gpus=0 actor 清除 HIP_VISIBLE_DEVICES。
4. 无测试配套改动：由于需要 ROCm 硬件，未增加 CI 测试，仅手动验证了 colocate 和 fully_async 两种训练路径。

关键文件：

- verl/workers/rollout/sglang_rollout/async_sglang_server.py（模块 rollout；类别 source；类型 dependency-wiring；符号 launch_server, launch_servers）：核心变更文件：默认 ROCm 下 attention 后端为 aiter，并重构 server actor runtime_env 为通过 platform 层注入环境变量。

- verl/trainer/constants_ppo.py（模块 trainer；类别 source；类型 dependency-wiring；符号 PPO_RAY_RUNTIME_ENV）：配置变更：在 PPO Ray runtime env 中新增 ROCm 特定的 RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO=0，修复 num_gpus=0 actor 无 GPU 可见性问题。

关键符号：launch_server, launch_servers, get_ppo_ray_runtime_env

关键源码片段

verl/workers/rollout/sglang_rollout/async_sglang_server.py

核心变更文件：默认 ROCm 下 attention 后端为 aiter，并重构 server actor runtime_env 为通过 platform 层注入环境变量。

```
# 文件：verl/workers/rollout/sglang_rollout/async_sglang_server.py
# 在 launch_server 方法中，当 attention_backend 未设置时，判断 ROCm 平台并默认使用 aiter
if attention_backend is None:
    if torch.version.hip is not None:
        attention_backend = "aiter" # ROCm 默认使用 AITER 加速
    elif version.parse(sglang.__version__) >= version.parse("0.5.12"):
        # FA3 CUDA-graph capture is broken on sglang>=0.5.12 (#22800);
        # default to flashinfer (users can opt into fa4 via engine_kwargs).
        attention_backend = "flashinfer"
    else:
        attention_backend = "fa3"

# 在 launch_servers 方法中，SGLang server actor 的 runtime_env 改为通过 platform 层获取
runtime_env={
    "env_vars": {
        # 使用 platform 提供的 NOSET 变量（ROCm 下为 HIP_VISIBLE_DEVICES 相关）
        **{var: "1" for var in get_platform().ray_noset_envvars()},
        # 使用 platform 提供的 rollout 环境变量（ROCm 下包含 SGLANG_USE_AITER=1）
        **get_platform().rollout_env_vars(),
    }
}
```

verl/trainer/constants_ppo.py

配置变更：在 PPO Ray runtime env 中新增 ROCm 特定的 RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO=0，修复 num_gpus=0 actor 无 GPU 可见性问题。

```
# 文件：verl/trainer/constants_ppo.py
# 在 _gb200_nccl_env 后新增 _rocm_ray_env，处理 ROCm 下 Ray 清除加速器可见性问题

# On ROCm, Ray 2.x force-clears accelerator visibility for num_gpus=0 actors
# (e.g. the SGLang server actor), leaving them unable to see any GPU. Disable
# that override so the actor keeps its HIP visibility. Scoped to ROCm to avoid
# changing Ray's default behavior on other platforms.
_rocm_ray_env = {}
# 仅在 ROCm 平台下注入该环境变量，避免影响 CUDA/NPU
```

```
if torch.version.hip is not None:
    _rocm_ray_env = {"RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO": "0"}

PPO_RAY_RUNTIME_ENV = {
    "env_vars": {
        # ... 其他通用环境变量 ...
        **_gb200_nccl_env,
        **_rocm_ray_env, # 合并到最终字典中
    },
}
```

评论区精华

核心讨论围绕 RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO 的必要性和作用域展开。审核者 wuxibin89 质疑：既然 SGLang server actor 已设置 RAY_EXPERIMENTAL_NOSET_CUDA_VISIBLE_DEVICES=1 使其可见所有设备，为何还需此变量。作者 xiaohong42 详细解释：在 ROCm 上，Ray 会强制将 num_gpus=0 actor 的 HIP_VISIBLE_DEVICES 置空，而空字符串会导致 HIP 隐式屏蔽所有 GPU（优先级高于 CUDA_VISIBLE_DEVICES），且该问题更早出现在 TaskRunner 导入 sglang 时。该设置需要作用在 job 级（PPO_RAY_RUNTIME_ENV）而非只针对 server actor。审核者最终认可并合入。

- RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO 必要性与作用域 (correctness): 作者通过详细实验数据和代码路径分析，说服审核者；最终保留该变量。
- 平台特定环境变量应移至 platform 层 (design): 已迁移，审核通过。

风险与影响

- 风险：
  1. 回归风险（低）：ROCm 特定变更均通过 torch.version.hip is not None 门控，不影响 CUDA/NPU 行为。但若未来 ROCm 环境改变或 Ray 版本更新，可能需重新评估。
  2. 性能风险（低）：将 attention backend 默认改为 aiter 可能不如原有 flashinfer/fa3 性能优化充分，但用户可通过 engine_kwargs 覆盖。
  3. 兼容性风险（低）：RAY_ACCEL_ENV_VAR_OVERRIDE_ON_ZERO 是 Ray 内部接口，未来可能不兼容，需关注 Ray 版本升级。
  4. 缺少测试覆盖：无单元测试或 CI，依赖手动验证。
- 影响：
  1. 用户影响：AMD ROCm 用户无需手动设置 SGLang 后端或环境变量即可运行 SGLang rollout，显著降低使用门槛。CUDA/NPU 用户无感知。
  2. 系统影响：新增了平台层与 SGLang rollout server 的集成模式，为未来其他平台扩展提供参考。
  3. 团队影响：无，代码量小且隔离良好。 - 风险标记：缺少测试覆盖，依赖 Ray 内部接口

关联脉络

- PR #6702 [hardware] feat: add ROCm/HIP platform backend (PlatformROCm): 本 PR 依赖 PlatformROCm 抽象层，使用其 rollout_env_vars() 和 ray_noset_envvars() 方法提供环境变量。
- PR #6565 [rollout, sglang] fix: migration sglang rollout to sglang 0.5.x: 可能是 SGLang rollout 相关的早期迁移工作，与本 PR 同属 SGLang rollout 演进。