

PR #7518 完整报告

verl-project/verl

[rollout, ci] fix: make agent-loop tests fully deterministic

合并时间: 2026-08-24 22:11

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7518>

执行摘要

- 一句话: 修复 agent-loop 测试不确定性, 实现全确定性采样与调度
- 推荐动作: 值得精读, 尤其关注确定性 request_id 生成与测试配置注入的设计。对于任何面临 CI flakiness 的团队, 该方案提供了从环境、采样、调度到 id 的完整确定性链路, 可作为参考模板。建议阅读时着重理解 `_make_request_id` 的开关设计, 以及 `AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG` 如何与 Hydra 配置系统衔接, 这两点是复用到其他测试或生产场景的关键。

功能与动机

Ascend CI 上 agent-loop 测试结果不稳定, 主要因为抽样、调度以及 request_id 生成天然随机。PR body 明确说明需“Configure deterministic sampling and scheduling, seed Python hashing, and generate stable request IDs from sample priority when full determinism is enabled”, 目的是让测试在完全确定模式下可重复, 避免 CI 偶发失败。关联的 `vllm_ascend` workflows 原先名为“Test the latest vLLM Rollout async with agent loop with seed 0(non deter)”, 本次重命名为“deterministic agent loop”, 印证了要根治不确定性的诉求。

实现拆解

实现拆解

1. 稳定 request id 生成 (`verl/experimental/agent_loop/tool_agent_loop.py`) - 新增静态方法 `_make_request_id(priority, full_determinism)`, 当 `full_determinism=True` 时返回 `f"det-{priority}"`, 否则回退到 `uuid4().hex`。 - 在 `run()` 中新增 `priority: int = 0` 关键字参数, 并将原先的 `request_id = uuid4().hex` 替换为 `self._make_request_id(priority, self.rollout_config.full_determinism)`。 - 这样 request id 不再随机, 而是与样本优先级挂钩, 保证在多样本并行时顺序稳定。
2. 测试配置可注入 (`tests/experimental/agent_loop/test_basic_agent_loop.py`) - 新增从环境变量 `AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG` 读取 JSON 字符串, 并通过 `OmegaConf.create()` 合并到 `init_config.actor_rollout_ref.rollout`。这样无需改测试代码即可通过环境变量覆盖温度、`top_p`、`seed` 等参数, 大幅提升灵活性。
3. CI workflows 启用全确定性 (`.github/workflows/vllm_ascend.yml`) - 将测试步骤重命名为“Test the latest vLLM Rollout async with deterministic agent loop”。 - 设置 `PYTHONHASHSEED: "0"` 固定哈希随机性。 - 注入 `AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG`

UT_CONFIG, 配置 temperature: 0.0、top_p: 1.0、top_k: -1、full_determinism: true、seed: 0、scheduling_policy: priority, 确保采样和调度均可复现。

整个改动形成闭环: 硬件层 (seed/ 环境变量) → 采样层 (温度 /top_p) → 调度层 (scheduling_policy) → request id 层 (priority 派生), 共同保证测试输出稳定。

关键文件:

- verl/experimental/agent_loop/tool_agent_loop.py (模块 Agent 循环; 类别 source; 类型 core-logic; 符号 _make_request_id, run): 核心源码改动, 新增 request id 确定性生成逻辑, 直接影响 agent-loop 运行行为。
- tests/experimental/agent_loop/test_basic_agent_loop.py (模块 Agent 循环; 类别 test; 类型 test-coverage): 测试配置注入, 使 CI 能通过环境变量覆盖确定性相关参数, 是确定性机制的验证入口。
- .github/workflows/vllm_ascend.yml (模块 CI 工作流; 类别 infra; 类型 infrastructure): CI 工作流启用全确定性环境, 是测试确定性落地的关键基础设施。

关键符号: _make_request_id, run

关键源码片段

verl/experimental/agent_loop/tool_agent_loop.py

核心源码改动, 新增 request id 确定性生成逻辑, 直接影响 agent-loop 运行行为。

```
# verl/experimental/agent_loop/tool_agent_loop.py

# 静态方法: 按确定性开关生成 request id
# 开启 full_determinism 时用 priority 生成稳定 id, 否则回退到随机 uuid
@staticmethod
def _make_request_id(priority: int, full_determinism: bool) -> str:
    # 确定性模式下: id 与样本优先级绑定, 保证多样本顺序可复现
    return f"det-{priority}" if full_determinism else uuid4().hex

@rollout_trace_op
async def run(self, sampling_params: dict[str, Any], priority: int = 0, **kwargs) -> AgentLoopOutput:
    messages = list(kwargs["raw_prompt"])

    # 提取多模态输入
    multi_modal_data = await self.process_multi_modal_info(messages)
    images = multi_modal_data.get("images")
    videos = multi_modal_data.get("videos")
    audios = multi_modal_data.get("audios")
    mm_processor_kwargs = self._get_mm_processor_kwargs(audios)

    metrics = {}
    # 关键: request_id 不再直接 uuid, 而是由 priority 与 full_determinism 决定
    request_id = self._make_request_id(priority, self.rollout_config.full_determinism)
    tools_kwargs = kwargs.get("tools_kwargs", {})
```

```

agent_data = AgentData(
    messages=messages,
    image_data=images,
    video_data=videos,
    audio_data=audios,
    mm_processor_kwargs=mm_processor_kwargs,
    metrics=metrics,
    request_id=request_id,
    tools_kwargs=tools_kwargs,
)
# ... 后续状态机循环逻辑不变

```

tests/experimental/agent_loop/test_basic_agent_loop.py

测试配置注入，使 CI 能通过环境变量覆盖确定性相关参数，是确定性机制的验证入口。

```

# tests/experimental/agent_loop/test_basic_agent_loop.py

# 允许通过环境变量注入确定性配置，便于 CI 控制温度 /seed/ 调度策略等
# 例如 AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG='{"temperature":0.0,...}'
if rollout_config := os.getenv("AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG"):
    # 用 OmegaConf 解析 JSON 字符串并合并到 rollout 配置，覆盖默认值
    init_config.actor_rollout_ref.rollout.merge_with(OmegaConf.create(rollout_config))
# 后续 agent_loop_manager 初始化与数据生成逻辑保持不变

```

.github/workflows/vllm_ascend.yml

CI 工作流启用全确定性环境，是测试确定性落地的关键基础设施。

```

# .github/workflows/vllm_ascend.yml

# 测试步骤：启用确定性 agent-loop
- name: Test the latest vLLM Rollout async with deterministic agent loop
  env:
    # 固定 Python 哈希种子，消除 dict/set 顺序随机性
    PYTHONHASHSEED: "0"
    # 注入确定性采样与调度配置
    AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG: |
      temperature: 0.0
      top_p: 1.0
      top_k: -1
      full_determinism: true
      seed: 0
      scheduling_policy: priority
  run: |
    export HCCL_HOST_SOCKET_PORT_RANGE=auto
    export HCCL_NPU_SOCKET_PORT_RANGE=auto
    # 后续运行测试逻辑

```

评论区精华

本 PR 只有 1 次 review, reviewer [wuxibin89](#) 直接 APPROVED, 无评论留言。因此没有公开的争论或设计讨论。不过根据代码可推断关键权衡: request_id 由随机 uuid 改为 deterministic 前缀, 测试中不同样本可通过 priority 区分, 但生产环境 (full_determinism=False) 仍保持 uuid 随机性, 避免影响正常并行追踪。改动虽小, 但实现上把确定性逻辑作为配置开关, 对未来开启确定性训练也有借鉴意义。

- 暂无高价值评论线程

风险与影响

- 风险:
 - request_id 唯一性风险: 当 full_determinism=True 时, request_id 变为 det-{priority}, 若同批中出现 priority 相同或跨 batch 复用, 可能产生重复 id, 理论上影响日志 / 追踪关联。由于该模式下仅用于测试且可控, 风险较低, 但若未来推广到生产需注意唯一性约束。
 - 调度策略依赖: CI 配置了 scheduling_policy: priority, 但调度逻辑是否与 priority 参数完全对齐未在 PR 中显式保证, 若调度器对 priority 的处理有边界情况 (如负数、超范围), 可能造成顺序不稳。
 - 环境变量注入链路: AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG 依赖 Hydra merge_with 正确处理, 若配置键路径变化 (如字段重命名) 可能导致 CI 静默失败, 需关注配置 schema 稳定性。
 - 测试覆盖有限: 测试仅覆盖 tool_agent 一种 agent, 其他 agent (如 basic) 未接入同一确定性机制, 可能存在未覆盖的 flakiness 来源。
 - 影响: 本 PR 主要影响 Ascend CI 的 agent-loop 测试稳定性, 直接降低 vllm_ascend 工作流的偶发失败率, 缩短维护者排查时间。对生产训练无影响, 因为 request_id 的确定性逻辑仅在 full_determinism 开启时生效, 默认仍为 uuid。改动的通用性较高, _make_request_id 与 AGENT_LOOP_TOOL_AGENT_ROLLOUT_CONFIG 注入方式可复用到其他 agent 类型, 未来若推广全确定性训练, 该实现可作为基础。对团队而言, 提升了 CI 可靠性与调试可复现性, 属于低风险高收益的稳定性改进。
 - 风险标记: request_id 可能与 priority 冲突, 调度策略依赖, 环境变量配置链路脆弱, 测试覆盖仅限 tool_agent

关联脉络

- PR #7491 [rollout] fix: preserve default AgentLoop extra fields: 同为 agent_loop 模块的修复, 涉及 AgentLoop 默认字段序列化问题, 与本 PR 同属 agent-loop 稳定性改进路线。
- PR #7513 [trainer, ckpt, cfg] feat: add config-driven checkpoint callback hook: 通过配置驱动行为, 与本 PR 用配置驱动确定性模式思路一致, 都是提升可配置性与可复现性。