

PR #6246 完整报告

verl-project/verl

[rollout,vllm] fix: include Ray job id in colocated weight-transfer IPC path

合并时间: 2026-05-06 16:27

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6246>

执行摘要

- 一句话: 修复 colocated vLLM 权重传输 IPC 路径跨作业冲突
- 推荐动作: 建议合并。这是一个精准的 bugfix, 设计轻量, 讨论中提及的 UUID 方案可以作为后续改进方向。阅读者应关注 IPC 路径隔离的设计模式, 可用于类似场景。

功能与动机

Issue #6233 报告了两个独立 verl 作业在同一主机上并发运行时, 由于 IPC socket 路径仅包含 `replica_rank` 和 `local_rank`, 导致第二个作业在 `update_weights` 时因 `Address already in use` 失败; 崩溃后残留的 socket 文件也会导致相同错误。PR body 明确目标是修复此碰撞问题。

实现拆解

1. vLLMHttpServer 传递 Ray job id 给子进程: 在 `vllm_async_server.py` 的 `__init__` 中, 紧接已有的 `VERL_REPLICA_RANK` 环境变量设置, 新增 `os.environ["VERL_RAY_JOB_ID"] = ray.get_runtime_context().get_job_id()`, 使得 vLLM worker 子进程能够继承该变量。
2. ServerAdapter (发送方) 使用 Ray job id 构造路径: 在 `vllm_rollout.py` 的 `ServerAdapter.__init__` 中, 通过 `ray.get_runtime_context().get_job_id()` 获取 job id, 并加入 `self.zmq_handle` 的路径字符串中, 确保发送方路径包含作业唯一标识。
3. vLLMColocateWorkerExtension (接收方) 读取环境变量构造匹配路径: 在 `utils.py` 的 `_get_zmq_handle` 方法中, 通过 `os.environ.get("VERL_RAY_JOB_ID", "0")` 读取 job id (若缺失则 fallback 为 "0"), 并加入返回的 IPC 路径中, 确保接收方与发送方路径一致。
4. 无用户可见的 API 或配置变更: `VERL_RAY_JOB_ID` 完全由内部自动设置和消费; 单作业运行行为不变。残留清理逻辑 (`bind()` 前的 `os.remove`) 继续有效, 且因路径隔离而更加安全。

关键文件:

- `verl/workers/rollout/vllm_rollout/utils.py` (模块 Rollout; 类别 source; 类型 core-logic; 符号 `_get_zmq_handle`): 接收方 (vLLM worker 子进程) 构造 IPC 路径的核心逻辑, 新增读取 `VERL_RAY_JOB_ID` 环境变量并拼入路径。
- `verl/workers/rollout/vllm_rollout/vllm_rollout.py` (模块 Rollout; 类别 source; 类型 core-logic; 符号 `ServerAdapter.init`): 发送方 (`ServerAdapter`, Ray actor) 构造 IPC 路径, 直接调用 `ray.get_runtime_context().get_job_id()` 获取 job id。

- verl/workers/rollout/vllm_rollout/vllm_async_server.py (模块 Rollout; 类别 source; 类型 core-logic; 符号 vLLMHttpServer.init) : vLLMHttpServer 在初始化时设置 VERL_RAY_JOB_ID 环境变量, 使其被子进程继承。

关键符号: _get_zmq_handle, vLLMHttpServer.init, ServerAdapter.init

关键源码片段

verl/workers/rollout/vllm_rollout/vllm_rollout.py

发送方 (ServerAdapter, Ray actor) 构造 IPC 路径, 直接调用 ray.get_runtime_context().get_job_id() 获取 job id。

```
# 位置: verl/workers/rollout/vllm_rollout/vllm_rollout.py
# ... 在 ServerAdapter.__init__ 中, 计算 local_rank 之后:
local_rank = self.rollout_rank % local_world_size
# 获取当前 Ray 作业的唯一 job id, 用于构造跨作业不冲突的 IPC 路径
job_id = ray.get_runtime_context().get_job_id()
# 构造 IPC 路径: 包含 job_id、replica_rank 和 local_rank
self.zmq_handle = f"ipc:///tmp/rl-colocate-zmq-{job_id}-replica-{self.replica_rank}-rank-{local_rank}.sock"
```

评论区精华

review 讨论较少。审核者 wuxibin89 在 PR 评论中提出: "We will change the replica_rank to an unique uuid." 表明考虑过使用 UUID 方案, 但当前 PR 使用了 Ray job id 作为更轻量的方案。gemini-code-assist[bot] 的自动审核无反馈。

- 使用 UUID 替代 Ray job id 的提议 (design): 未采纳, 当前 PR 采用更轻量的 Ray job id 方案, 无需新增公共 env var 契约。

风险与影响

- 风险: 低风险。变更仅涉及 IPC 路径字符串构造, 不修改协议或逻辑。"0" fallback 确保接收方缺失 env var 时仍能快速失败 (路径不匹配), 不会静默污染其他作业。单作业运行行为完全不变。需确保 Ray job id 在跨节点场景下相同 (Ray 保证)。
- 影响: 正面影响: 解决多作业 / 崩溃残留导致的启动失败问题, 提高多租户共享节点场景的稳定性和可靠性。无用户 API 变更, 无回退需求。影响范围限于 colocated vLLM rollout 模式。
- 风险标记: 核心路径变更, 低风险

关联脉络

- PR #6062 [rollout] fix: use replica_rank to avoid IPC path collisions when multiple replicas share a node: 本 PR 在原有 replica_rank/local_rank 方案基础上, 新增 Ray job id 维度, 解决跨作业冲突。