PR #6861 完整报告

verl-project/verl

[vllm] fix: crash in start_profile/stop_profile on non-master nodes when nnodes > 1

合并时间：2026-07-01 05:50

原文链接：http://prhub.com.cn/verl-project/verl/pull/6861

执行摘要

- 一句话：修复多节点 profile 崩溃
- 推荐动作：值得合并，修复明确，风险低，测试完备。对于依赖于多节点 profile 的团队，此 PR 修复了严重的运行时崩溃问题。

功能与动机

PR body 明确指出：当 replica 跨多节点（nnodes > 1）时，只有 node_rank == 0 运行 run_server() 初始化 self.engine，其他节点运行 run_headless() 使 self.engine 为 None。但 start_profile() 和 stop_profile() 通过 asyncio.gather 广播到所有 replica，导致非 master 节点访问 self.engine 时抛出 AttributeError。

实现拆解

1. 核心逻辑修复：在 verl/workers/rollout/vllm_rollout/vllm_async_server.py 的 start_profile 和 stop_profile 方法开头各增加一句 if self.node_rank != 0: return，在调用 self.engine 之前提前退出。这与文件中已有的 clear_kv_cache、release_kv_cache、resume_kv_cache 等方法的 node_rank 守卫模式保持一致。
2. 补充单元测试：在 tests/utils/test_server_profiler.py 中新增 test_vllm_start_stop_profile_non_master_node 异步测试用例。该测试设置 mock_self.node_rank = 1，分别调用 start_profile 和 stop_profile，并验证 mock_engine.start_profile 和 mock_engine.stop_profile 均未被调用（assert_not_called）。
3. 调整现有测试：在已有的 test_vllm_start_stop_profile 用例中，为 mock_self 显式设置 mock_self.node_rank = 0，以确保该测试在 master 节点场景下依然正确。

关键文件：

- verl/workers/rollout/vllm_rollout/vllm_async_server.py（模块 vLLM 卷出；类别 source；类型 core-logic；符号 start_profile, stop_profile）：核心修复文件，在 start_profile 和 stop_profile 中添加 node_rank 守卫，避免非 master 节点崩溃。
- tests/utils/test_server_profiler.py（模块 性能分析器；类别 test；类型 test-coverage；符号 test_vllm_start_stop_profile_non_master_node, test_vllm_start_stop_profile）：新增测试覆盖非 master 节点场景，并修复原有测试以正确设置 node_rank。

关键符号：start_profile, stop_profile, test_vllm_start_stop_profile_non_master_node

关键源码片段

verl/workers/rollout/vllm_rollout/vllm_async_server.py

核心修复文件，在 start_profile 和 stop_profile 中添加 node_rank 守卫，避免非 master 节点崩溃。

```
# verl/workers/rollout/vllm_rollout/vllm_async_server.py
# 修改前：start_profile 和 stop_profile 没有 node_rank 守卫，
# 在 nnodes > 1 时，非 master 节点的 self.engine 为 None，引发 AttributeError。
# 修改后：与 clear_kv_cache / release_kv_cache / resume_kv_cache 保持一致，
# 非 master 节点提前 return，不再调用 engine 方法。

async def start_profile(self, **kwargs):
    if self.node_rank != 0:
        return
    if (
        self.profiler_controller.check_enable()
        and self.profiler_controller.check_this_rank()
        and self.profiler_controller.is_discrete_mode()
    ):
        await self.engine.start_profile(**kwargs)

async def stop_profile(self):
    if self.node_rank != 0:
        return
    if (
        self.profiler_controller.check_enable()
        and self.profiler_controller.check_this_rank()
        and self.profiler_controller.is_discrete_mode()
    ):
        await self.engine.stop_profile()
```

tests/utils/test_server_profiler.py

新增测试覆盖非 master 节点场景，并修复原有测试以正确设置 node_rank。

```
# tests/utils/test_server_profiler.py
# 新增测试：验证非 master 节点不会调用 engine 的 profile 方法
async def test_vllm_start_stop_profile_non_master_node(self):
    try:
        from verl.workers.rollout.vllm_rollout.vllm_async_server import vLLMHttpServer
    except ImportError:
        self.skipTest("vllm or dependencies not installed")
        return

    mock_profiler = MagicMock()
    mock_profiler.check_enable.return_value = True
    mock_profiler.check_this_rank.return_value = True
    mock_profiler.is_discrete_mode.return_value = True
```

```
    mock_engine = AsyncMock()

    mock_self = MagicMock()
    mock_self.node_rank = 1 # non-master node, should skip
    mock_self.profiler_controller = mock_profiler
    mock_self.engine = mock_engine

    await vLLMHttpServer.start_profile(mock_self)
    mock_engine.start_profile.assert_not_called()

    await vLLMHttpServer.stop_profile(mock_self)
    mock_engine.stop_profile.assert_not_called()

# 同时，在已有的 test_vllm_start_stop_profile 中增加了 mock_self.node_rank = 0
```

评论区精华

主要讨论围绕测试修复展开。审核者 @Luosuu 要求作者修复失败的测试 test_vllm_start_stop_profile。作者 @kyle-zhangchi 在后续提交中增加了新测试用例并修正了原有测试（添加 mock_self.node_rank = 0）。后续 CI 通过后，@kyle-zhangchi 指出失败测试与本次变更无关，@Luosuu 批准合并。

- 测试修复 (testing): 测试已修复并通过 CI。

风险与影响

- 风险：风险极低。变更只添加了 if guard，不改变已有控制流，且与项目中其他类似方法（如 clear_kv_cache、release_kv_cache）的模式一致。测试覆盖了 master 和 non-master 两种场景。
- 影响：仅影响多节点部署（nnodes > 1）下 vLLM rollout 的 profile 功能。修复后非 master 节点不会再因访问 None 的 engine 而崩溃，profile 操作仅在 master 节点执行。
- 风险标记：修复简单，风险低

关联脉络

- PR #4320 相关 PR 参考（PR body 中提及的搜索链接）：作者在 PR body 中引用此 PR 作为类似 PR 搜索的结果。