PR #6620 完整报告

verl-project/verl

[vllm] fix: use data-parallel rank in vLLM ZMQ handles

合并时间：2026-06-08 16:13

原文链接：http://prhub.com.cn/verl-project/verl/pull/6620

执行摘要

- 一句话：修复 vLLM ZMQ handle 在 DP>1 时的 rank 冲突
- 推荐动作：建议分布式训练工程师精读该 PR，尤其是 rank 计算逻辑。_resolve_vllm_weight_sync_local_rank 的 fallback 路径值得关注，建议未来增加日志以降低调试成本。

功能与动机

Issue #6615 报告在 DP>1 且 EP 启用时，多个 vLLM worker 共享受同名称的 socket 导致冲突。sender 侧使用 rollout-local rank 构造 handle，而 receiver 侧使用 vLLM worker 的 self.local_rank，该 rank 是 TP 局部的，导致不同 DP worker 的 rank 相同，引发电气通信错误。

实现拆解

1. 分析问题：识别出 sender 与 receiver 使用不同的 rank 计算方式，导致 DP>1 时 socket 冲突。
2. 引入 rank 解析函数：在 verl/workers/rollout/vllm_rollout/utils.py 中新增 _resolve_vllm_weight_sync_local_rank。该函数接受 worker 本地 rank 和并行配置，通过 data_parallel_rank_local * tensor_parallel_size + tensor_parallel_rank 计算出与 sender 一致的 rank。如果 DP 未启用，则返回原 rank 保持兼容。
3. 修改 ZMQ handle 构造：在 vLLMColocateWorkerExtension._get_zmq_handle 中，调用解析函数替换直接使用 self.local_rank 的行为。
4. 添加单元测试：在 test_vllm_cli_args_on_cpu.py 中新增 TestVllmColocateZmqHandle 类，覆盖 DP 共置、单 DP、全局 DP fallback 以及 ZMQ handle 集成测试。

关键文件：

- verl/workers/rollout/vllm_rollout/utils.py（模块 Rollout；类别 source；类型 core-logic；符号 _resolve_vllm_weight_sync_local_rank, _get_zmq_handle）：核心源码，引入新的 rank 解析函数并修改 ZMQ handle 构造。
- tests/workers/rollout/test_vllm_cli_args_on_cpu.py（模块 测试；类别 test；类型 test-coverage；符号 TestVllmColocateZmqHandle, test_dp_local_rank_offsets_tensor_parallel_rank, test_single_dp_keeps_local_rank, test_uses_global_dp_rank_when_local_rank_is_unset）：新增测试类覆盖各种 DP 配置场

景，确保 rank 解析正确。

关键符号：_resolve_vllm_weight_sync_local_rank, _get_zmq_handle

关键源码片段

tests/workers/rollout/test_vllm_cli_args_on_cpu.py

新增测试类覆盖各种 DP 配置场景，确保 rank 解析正确。

```
class TestVllmColocateZmqHandle:
    def test_dp_local_rank_offsets_tensor_parallel_rank(self):
        """DP workers 在同一节点必须使用不同 socket。"""
        # 模拟 TP=2, DP=4, local DP size=2, local DP rank=1
        parallel_config = SimpleNamespace(
            tensor_parallel_size=2,
            data_parallel_size=4,
            data_parallel_size_local=2,
            data_parallel_rank_local=1,
        )
        # 预期：1 * 2 + (1 % 2) = 2 + 1 = 3
        assert _resolve_vllm_weight_sync_local_rank(1, parallel_config) == 3
        # TP rank = 3 % 2 = 1
        assert _resolve_vllm_weight_sync_local_rank(3, parallel_config) == 3

    def test_single_dp_keeps_local_rank(self):
        """单 DP 场景保持原始 local rank 不变。"""
        parallel_config = SimpleNamespace(
            tensor_parallel_size=2, data_parallel_size=1,
            data_parallel_size_local=1, data_parallel_rank_local=0,
        )
        assert _resolve_vllm_weight_sync_local_rank(1, parallel_config) == 1

    def test_uses_global_dp_rank_when_local_rank_is_unset(self):
        """当 local DP rank 未设置时，从 global DP rank 计算 fallback。"""
        # dp_local_rank = None, dp_rank=3, dp_local_size=2
        # dp_local_rank fallback = 3 % 2 = 1
        # 结果：1 * 2 + (0 % 2) = 2 + 0 = 2
        parallel_config = SimpleNamespace(
            tensor_parallel_size=2, data_parallel_size=4,
            data_parallel_size_local=2, data_parallel_rank_local=None,
            data_parallel_rank=3,
        )
        assert _resolve_vllm_weight_sync_local_rank(0, parallel_config) == 2

    def test_zmq_handle_uses_resolved_dp_rank(self, monkeypatch):
        """验证 _get_zmq_handle 使用解析后的 rank 构造 socket 路径。"""
        parallel_config = SimpleNamespace(
            tensor_parallel_size=2, data_parallel_size=4,
            data_parallel_size_local=2, data_parallel_rank_local=1,
```

```
)
worker = SimpleNamespace(
    local_rank=1,
    model_runner=SimpleNamespace(
        vllm_config=SimpleNamespace(parallel_config=parallel_config),
    ),
)
monkeypatch.setenv("VERL_REPLICA_RANK", "2")
monkeypatch.setenv("VERL_RAY_JOB_ID", "job-123")
handle = vLLMColocateWorkerExtension._get_zmq_handle(worker)
# DP rank 解析为 3，所以 rank-3.sock
expected = "ipc:///tmp/rl-colocate-zmq-job-123-replica-2-rank-3.sock"
assert handle == expected
```

评论区精华

gemini-code-assist[bot] 在 review 中指出，当 dp_size > 1 但 dp_local_rank 无法解析时，_resolve_vllm_weight_sync_local_rank 静默回退到 worker_local_rank，这可能导致 socket 冲突，建议添加 warning 日志。该建议未被采纳，PR 已合并时未添加 warning。

- silent fallback 时添加 warning 日志 (design): 未采纳，PR 合并时未添加 warning。

风险与影响

- 风险：主要风险是 _resolve_vllm_weight_sync_local_rank 中当 dp_local_rank 无法解析（为 None）时，函数会 fallback 到原始 worker_local_rank，这在 DP 启用但 rank 信息缺失的场景下仍可能造成 socket 冲突。建议未来添加 warning 日志以辅助调试。此外，该函数依赖 parallel_config 的属性名称，若 vLLM 版本升级改变属性名，可能导致回退行为。但当前实现向后兼容单 DP 场景。
- 影响：对使用 colocated vLLM 且 DP>1 的用户是必要的修复，使权重同步建立正确连接；单 DP 场景向后兼容，无需配置变更。影响范围限于分布式训练中启用 vLLM 共置部署的用户。
- 风险标记：缺失 fallback 警告，配置依赖 vLLM 内部属性

关联脉络

- 暂无明显关联 PR