

# PR #7488 完整报告

verl-project/verl

[ckpt] fix: hccl error when update weights for npu

合并时间: 2026-08-20 20:06

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7488>

## 执行摘要

- 一句话: 回退异步广播, 修复 NPU 权重更新时 hccl error
- 推荐动作: 值得快速浏览, 作为 "性能优化在硬件专属路径上引入回归后被回退" 的典型案  
例: 提醒在 HCCL 等硬件专属通信路径上做并行化 / 异步化改造时, 必须有对应 NPU 环境  
的测试覆盖。无需精读, 技术含量有限, 但可在 release note 中记录此次回退以避免他人  
重复踩坑。

## 功能与动机

PR body 说明: "This reverts commit 352f76f59ff014c6b423eed1a21a7881be1a0b46.  
#7205 will cause hccl error when update weights for npu"。即此前 #7205 引入的异步化  
改造在 NPU 环境下触发 HCCL 通信错误, 本 PR 目标是恢复 #7205 之前已知可用的同步广播  
行为, 保证 NPU 上权重更新流程恢复正常。

## 实现拆解

变更全部集中在 `verl/checkpoint_engine/hccl_checkpoint_engine.py` 的  
`BroadcastOperation` 类, 共 4 步:

1. 移除 `asyncio` 依赖: 删除文件顶部 `import asyncio`, 改造后不再需要 `event loop`。
2. 构造函数改同步执行: `__init__` 中删除 `loop = asyncio.get_running_loop()` 与 `self._task = loop.run_in_executor(None, self._run)` 两行, 改为直接调用 `self._run()`。这是核心回退——广播从后台线程异步执行变为调用线程内同步执行, 规避 NPU 下 HCCL 在线程池中触发错误的问题。
3. 收尾逻辑简化: `wait_for_complete()` 删除 `await self._task`, 直接 `return self.metadata`; 方法签名仍保留 `async def`, 避免改动调用方 API 契约。
4. 无配套测试 / 配置变更: 本次未新增测试文件, 依赖现有 Ascend CI 工作流 (例如 `e2e_ppo_trainer_megatron_sglang_ascend.yml`) 做 NPU 环境验证。

关键点: 回退后 `self.pyhccl.broadcast(...)` 在 `__init__` 返回前已同步发起, `wait_for_complete()` 返回的 `metadata` 必然已更新, 正确性有保障; 代价是失去异步执行带来的潜在重叠收益。

关键文件:

- verl/checkpoint\_engine/hccl\_checkpoint\_engine.py (模块 检查点引擎; 类别 source; 类型 core-logic; 符号 BroadcastOperation.init, BroadcastOperation.\_run, BroadcastOperation.wait\_for\_complete) : 唯一变更文件, 包含整次回退的核心逻辑: 移除 asyncio 依赖, 将 HCCL 广播从 executor 线程异步执行改回同步执行。

关键符号: BroadcastOperation.init, BroadcastOperation.\_run,  
BroadcastOperation.wait\_for\_complete

## 关键源码片段

### verl/checkpoint\_engine/hccl\_checkpoint\_engine.py

唯一变更文件, 包含整次回退的核心逻辑: 移除 asyncio 依赖, 将 HCCL 广播从 executor 线程异步执行改回同步执行。

```
class BroadcastOperation:
    """HCCL 广播操作 (回退后为同步执行) 。
```

Args:

- rank (int): 当前进程的 rank。
- process\_group (StatelessProcessGroup | str): HCCL 进程组。
- bucket (torch.Tensor): 待广播的 tensor。
- metadata (dict[str, TensorMeta]): tensor 的元信息。
- socket (zmq.Socket): 与 master 通信的 zeromq socket。
- topic (str): 订阅的 topic。

```
"""

def __init__(
    self,
    rank: int,
    process_group: StatelessProcessGroup | str,
    bucket: torch.Tensor,
    metadata: dict[str, TensorMeta],
    socket: zmq.Socket,
    topic: str,
) -> None:
    self.rank = rank
    self.pyhccl = process_group
    self.bucket = bucket
    self.metadata = metadata
    self.socket = socket
    self.topic = topic

    # 关键回退: 不再用 asyncio executor 线程跑 HCCL,
    # 改为当前线程内同步执行, 避免 NPU 上更新权重时的 hccl error
    self._run()

def _run(self):
    # 第一步: 通过 zeromq PUB/SUB 同步 tensor 元信息
    if self.rank == 0:
```

```

        self.socket.send_string(self.topic, flags=zmq.SNDMORE)
        self.socket.send_pyobj(self.metadata)
    else:
        self.socket.recv_string()
        self.metadata = self.socket.recv_pyobj()

    # 第二步：通过 HCCL 广播 tensor 数据本体
    self.pyhccl.broadcast(self.bucket, src=0)

    async def wait_for_complete(self) -> dict[str, TensorMeta]:
        """等待广播完成。

        注意：HCCL 广播已在 __init__ 中同步执行，
        此处仅返回广播后的元信息，不保证 HCCL kernel 已真正结束。
        """

        return self.metadata

```

## 评论区精华

本 PR 没有任何技术性 review 讨论：唯一的评论来自 CLAassistant 机器人，提示 "r1chardf1d0 seems not to be a GitHub user"，要求作者把 commit 邮箱关联到 GitHub 账号才能签署 CLA，属于流程提醒。维护者 wucong25 直接 APPROVED 且未留评语，说明回退方向明确、无需额外讨论。评论区没有展开 "为何不在 NPU 上修复线程执行根因" 的技术辩论，本次选择以最小代价回退规避。

- CLA 签署状态提醒 (other): 该提醒属于流程通知，不涉及技术内容；PR 最终由维护者 wucong25 批准并合入，说明 CLA 问题未阻塞合入流程。

## 风险与影响

- 风险：
  1. 性能回退：广播从异步变同步，原先 run\_in\_executor 带来的执行重叠收益消失；权重规模大的场景下，广播期间无法与其他计算 / 通信重叠，可能拉长权重同步耗时。
  2. 语义偏差：wait\_for\_complete() 仍是 async def 但函数体无 await，会立即返回；类 docstring 仍写着 "Async broadcast operation with HCCL in separate thread"，与实际同步行为不符，容易误导后续维护者。
  3. 根因未除：HCCL 在 executor 线程中触发 NPU hccl error 的根因没有定位，本次只是恢复已知可用行为；未来若再次尝试异步化，需先在 NPU 上复现并定位线程 / event loop 冲突。
  4. 测试覆盖不足：无新增测试，仅依赖 Ascend CI 做 NPU 环境验证，如果 CI 未覆盖到该更新权重路径，回归风险需人工确认。
  5. 影响范围受限：仅 HCCLCheckpointEngine 路径 (Ascend NPU)，CUDA 侧 nccl\_checkpoint\_engine.py 不受影响。- 影响：对 Ascend NPU 用户是直接的正确性修复：更新权重时不再触发 hccl error，权重同步流程恢复可用。对系统整体影响面窄，改动单文件、行为回退到 #7205 之前的已知状态，符合 "最小变更" 原则。对团队而言，这是一次低风险的紧急修复，但遗留了性能优化与 NPU 线程安全的未来工作。- 风险

标记: NPU 专属路径变更, 性能回退风险, 缺少测试覆盖, 根因未定位

## 关联脉络

- PR #7205 commit 352f76f (正文引用, 具体标题未知): PR body 明确说明 #7205 会在 NPU 更新权重时引发 hccl error, 本 PR 正是对该改动 (commit 352f76f) 的完整回退。
- PR #7291 [ckpt] feat: Node-local multi-sender broadcast in NCCL checkpoint engine: 同一 checkpoint engine 模块下的权重广播功能演进, 与 HCCL 广播路径同属权重同步体系, 本回退是该演进方向上的修正性变更。