PR #6463 完整报告

verl-project/verl

[fsdp] fix: device mismatch between fsdp2 offload and weights transfer

合并时间：2026-05-25 19:31

原文链接：http://prhub.com.cn/verl-project/verl/pull/6463

执行摘要

- 一句话：修复 FSDP2 CPUOffloadPolicy 下 state_dict 设备不匹配崩溃
- 推荐动作：建议合并，同时尽快修复 save_checkpoint 的类似问题以保持代码一致性和避免后续崩溃。可参考本 PR 的守卫模式。

功能与动机

在 FSDP2 + CPUOffloadPolicy 配置下，actor 使用完整权重训练时会调用 update_weights() 进行权重同步，导致 get_per_tensor_param() 内的 state_dict() 崩溃。根因是手动调用 model.to(device) 与 CPUOffloadPolicy 自动管理的设备放置冲突，触发 RuntimeError: Attempted to set the storage of a tensor on device 'cpu' to a storage on different device 'cuda:0'。issue #5995 中 mikequan0425 提出直接跳过 model.to(device) 的解决方案。

实现拆解

1. 在 FSDPEngine.__init__中新增属性 _uses_fsdp2_cpu_offload_policy = False。
2. 在 FSDPEngine._build_fsdp_module 的 fsdp2 分支中，当配置 CPUOffloadPolicy 时将该属性设为 True。
3. 修改 get_per_tensor_param 方法：仅在 _uses_fsdp2_cpu_offload_policy 为 False 时调用 load_fsdp_model_to_gpu，跳过手动移动。
4. 新增回归测试文件 tests/special_distributed/test_fsdp2_cpu_offload_state_dict.py，使用极小 Qwen2 模型模拟 FSDP2 + CPUOffloadPolicy，验证修复后 state_dict 成功且 DTensor 材质化仍生成 GPU 张量，同时保持前修复崩溃的探测（仅信息性）。
5. 在 tests/special_distributed/run_all.sh 中添加新测试的 torchrun 命令，确保分布式测试中执行。

关键文件：

- verl/workers/engine/fsdp/transformer_impl.py（模块 FSDP 引擎；类别 source；类型 core-logic）：核心修改：新增 _uses_fsdp2_cpu_offload_policy 标志并在 get_per_tensor_param 中跳过手动模型移动
- tests/special_distributed/test_fsdp2_cpu_offload_state_dict.py（模块 FSDP 测试；类别 test；类型 test-coverage；符号 _build_fsdp2_cpu_offload_module, _assert_fixed_path_succeeds, _probe_pre_fix_crash, main）：新增回归测试，覆盖

FSDP2 + CPUOffloadPolicy 下 state_dict 的正确路径和探测崩溃路径

- tests/special_distributed/run_all.sh（模块 测试脚本；类别 test；类型 test-coverage）：添加新测试的运行命令到分布式测试集合

关键符号：FSDPEngine.get_per_tensor_param, FSDPEngine.init, FSDPEngine._build_fsdp_module, _assert_fixed_path_succeeds, _probe_pre_fix_crash

关键源码片段

verl/workers/engine/fsdp/transformer_impl.py

核心修改：新增 _uses_fsdp2_cpu_offload_policy 标志并在 get_per_tensor_param 中跳过手动模型移动

```
def get_per_tensor_param(self, layered_summon=False, base_sync_done=False, **kwargs):
    log_gpu_memory_usage("Before load_fsdp_model_to_gpu", logger=logger)
    # FSDP2 CPUOffloadPolicy 自行管理 CPU <-> GPU 放置，手动调用
    # model.to(device) 会让模块处于半迁移状态，继而导致下方的
    # state_dict() 崩溃 (#5995)。后续的 per-DTensor .to(device).full_tensor()
    # 仍能生成 GPU 张量，因此跳过手动移动。
    if not self._uses_fsdp2_cpu_offload_policy:
        load_fsdp_model_to_gpu(self.module)
    log_gpu_memory_usage("After load_fsdp_model_to_gpu", logger=logger)
    # 后续代码：调用 self.module.state_dict() 等保持不变
```

tests/special_distributed/test_fsdp2_cpu_offload_state_dict.py

新增回归测试，覆盖 FSDP2 + CPUOffloadPolicy 下 state_dict 的正确路径和探测崩溃路径

```
def _assert_fixed_path_succeeds(device_mesh, rank):
    """Replay FSDPEngine.get_per_tensor_param's post-fix sequence."""
    module = _build_fsdp2_cpu_offload_module(device_mesh)
    # 修复后：不调用 load_fsdp_model_to_gpu，直接 state_dict
    state_dict = module.state_dict()
    assert len(state_dict) > 0, "expected a populated state dict"
    # 验证 DTensor 材质化仍然产生 GPU 张量
    device = get_device_id()
    materialised = False
    for name, param in state_dict.items():
        if isinstance(param, DTensor):
            full = param.to(device, non_blocking=True).full_tensor()
            assert full.device.type == get_device_name(), (
                f"{name}: full_tensor() yielded {full.device.type}, expected {get_device_name()}"
            )
            materialised = True
            break
    assert materialised, "did not encounter any DTensor in state_dict; FSDP2 sharding may not be active"
    if rank == 0:
        print("fixed path: state_dict() + DTensor materialisation succeeded")
```

评论区精华

Gemini-code-assist 自动审查指出 save_checkpoint 方法也存在同样的无条件 load_fsdp_model_to_gpu 调用，可能遭遇相同崩溃，建议加入守卫。该建议未在本次 PR 中处理，但 PR 仍被批准合并。

- save_checkpoint 也存在相同问题 (correctness): 未在本 PR 中处理，作者未回应，但 PR 被批准合并，留待后续修复。

风险与影响

- 风险：核心风险：对非 CPUOffloadPolicy 场景无影响。save_checkpoint 路径仍有相同 bug，若用户同时启用 CPUOffloadPolicy 和检查点保存，可能崩溃。新增测试依赖分布式环境（2 GPU），已加入 run_all.sh，但可能未在 CI 的快速测试中覆盖。变更逻辑简单，回归风险低。
- 影响：影响范围：仅影响使用 FSDP2 + CPUOffloadPolicy + 完整权重训练的用户（即 issue 描述的场景）。修复后权重同步正常，训练不再中断。不启用 CPUOffloadPolicy 的用户完全不受影响。对系统稳定性有积极影响。
- 风险标记：save_checkpoint 未修复，仅覆盖 FSDP2 CPUOffloadPolicy 路径，测试依赖分布式环境

关联脉络

- PR #5995 [Bug] FSDP2 CPUOffloadPolicy + state_dict() crashes with device mismatch during update_weights: 本 PR 直接修复的根因 issue，其讨论中提出了解决方案