PR #7103 完整报告

verl-project/verl

[veomni, fsdp] fix: skip manual model.to(device) under FSDP2 CPU offload

合并时间：2026-07-21 15:17

原文链接：http://prhub.com.cn/verl-project/verl/pull/7103

执行摘要

- 一句话：修复 VeOmni 引擎在 FSDP2 卸载下的崩溃
- 推荐动作：值得精读，特别是对于使用 VeOmni 引擎且启用 FSDP2 CPU offload 的团队。PR 设计清晰，与 #6604 的 FSDP 引擎修复保持了一致性，同时代码中注释详实，解释了为何跳过 load_veomni_model_to_gpu 是安全的。

功能与动机

当 VeOmni 引擎使用 enable_fsdp_offload=True 时，FSDP2 的 CPUOffloadPolicy 会自动管理参数在 CPU 与加速器之间的放置。手动调用 model.to(device) 会导致参数本地存储（CPU）与元数据设备（加速器）不一致，进而触发 PyTorch 的 RuntimeError: Attempted to set the storage of a tensor on device "cpu" to a storage on different device "npu:0"。此错误已在 FSDP 引擎（#6604）中修复，但 VeOmni 引擎有相同的代码路径却未被覆盖，因此需要同样的守卫。

实现拆解

1. 初始化阶段 (__init__)：在 verl/workers/engine/veomni/transformer_impl.py 中，新增 self._uses_fsdp2_cpu_offload_policy = self.engine_config.enable_fsdp_offload，记录是否启用了 FSDP2 CPU offload。
2. save_checkpoint 守卫：将原有的无条件 load_veomni_model_to_gpu(self.module) 改为条件调用，仅当 _uses_fsdp2_cpu_offload_policy 为 False 时才执行。
3. load_checkpoint 守卫：同样为 load_veomni_model_to_gpu(self.module) 添加 _uses_fsdp2_cpu_offload_policy 条件检查。
4. get_per_tensor_param 守卫：为 load_veomni_model_to_gpu(self.module) 添加相同条件检查，因为该函数中后续的 param_generator() 已能自动处理设备移动，跳过显式移动是安全的。

关键文件：

- verl/workers/engine/veomni/transformer_impl.py（模块 引擎；类别 source；类型 core-logic；符号 VeOmniEngine.init, VeOmniEngine.save_checkpoint, VeOmniEngine.load_checkpoint, VeOmniEngine.get_per_tensor_param）：唯一变更文件，在 __init__ 中新增 _uses_fsdp2_cpu_offload_policy 标志，并在 save_checkpoint、load_checkpoint、get_per_tensor_param 三个方法中为 load_veomni_model_to_gpu 调

用添加守卫条件。

关键符号：VeOmniEngine.init, VeOmniEngine.save_checkpoint, VeOmniEngine.load_checkpoint, VeOmniEngine.get_per_tensor_param

关键源码片段

verl/workers/engine/veomni/transformer_impl.py

唯一变更文件，在 __init__ 中新增 _uses_fsdp2_cpu_offload_policy 标志，并在 save_checkpoint、load_checkpoint、get_per_tensor_param 三个方法中为 load_veomni_model_to_gpu 调用添加守卫条件。

```
# verl/workers/engine/veomni/transformer_impl.py

# 在 __init__ 中，第 152 行新增标志
self._uses_fsdp2_cpu_offload_policy = self.engine_config.enable_fsdp_offload
# 当 enable_fsdp_offload=True 时，FSDP2 使用 CPUOffloadPolicy，
# 手动调用 model.to(device) 会导致 DTensor 存储设备不匹配（#5995 / #6604）

# save_checkpoint 方法中，第 553 行
origin_module_device = next(self.module.parameters()).device.type
if (self._is_offload_param or origin_module_device == "cpu") and not getattr(
    self, "_uses_fsdp2_cpu_offload_policy", False
):
    load_veomni_model_to_gpu(self.module)
# 如果启用了 FSDP2 CPU offload，跳过显式加载到 GPU

# load_checkpoint 方法中，第 569 行
if self._is_offload_param and not getattr(self, "_uses_fsdp2_cpu_offload_policy", False):
    load_veomni_model_to_gpu(self.module)
# 同样，仅当未启用 FSDP2 CPU offload 时才移动模型到 GPU

# get_per_tensor_param 方法中，第 587 行
if not getattr(self, "_uses_fsdp2_cpu_offload_policy", False):
    load_veomni_model_to_gpu(self.module)
# 跳过整体模型移动，因为后续的 param_generator() 中的 full_tensor()
# 仍能返回加速器张量，该移动在 CPU offload 下是多余的
```

评论区精华

Gemini Code Assist 的自动审查未提出具体问题，仅确认 PR 引入了守卫条件以解决设备不匹配崩溃。wuxibin89 批准了该 PR。无其他讨论。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险是当 enable_fsdp_offload=True 时，跳过 load_veomni_model_to_gpu 可能导致后续 state_dict() 或其他操作在预期参数在 GPU 上时失败。但 PR 说明指出

param_generator() 中的 full_tensor() 调用仍能返回加速器张量，且 FSDP2CPUOffloadPolicy 会自行管理设备。此外，本变更仅影响 VeOmni 引擎，且 _uses_fsdp2_cpu_offload_policy 默认 False，不会影响原有非 offload 路径。潜在的回归风险较低。

- 影响：直接影响：修复了 VeOmni 引擎在 FSDP2 CPU offload 模式下 checkpoint 保存 / 加载和权重导出的崩溃问题。影响范围仅限于 VeOmni 引擎，不涉及 FSDP、Megatron 等其他引擎。团队可预期在 Ascend NPU 等硬件上使用 enable_fsdp_offload=True 的 VeOmni 训练流程不再因设备不匹配而中断。
- 风险标记：仅 VeOmni 引擎受影响，缺少自动化回归测试

关联脉络

- PR #6604 [veomni, fsdp] fix: skip manual model.to(device) under FSDP2 CPU offload: 本 PR 是 #6604 的延续，将 FSDP 引擎中的同一修复应用到 VeOmni 引擎。
- PR #5995 [Bug] FSDP2 CPUOffloadPolicy + state_dict() crashes with device mismatch during update_weights (non-LoRA full-weight training): 根因 issue，描述了 FSDP2 CPU offload 下 state_dict() 的设备不匹配崩溃。
- PR #6463 [fsdp] fix: device mismatch between fsdp2 offload and weights transfer: 早期尝试修复同一问题的 PR，但未覆盖 VeOmni。
- PR #7005 [fsdp] fix: skip the whole-shard staging round trip in FSDP2 weight export: 后续优化 PR，在 FSDP2 权重导出中跳过不必要的 GPU 暂存，与本 PR 的守卫逻辑互补。