

PR #7539 完整报告

verl-project/verl

[ray] fix: skip unused TensorDict consolidation in NumPy DataProto serialization

合并时间: 2026-08-25 12:09

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7539>

执行摘要

- 一句话: 修复 NumPy 序列化中未使用的大块 TensorDict consolidation 分配
- 推荐动作: 此 PR 值得精读, 因为它揭示了序列化路径中一个隐蔽的性能浪费点, 并且是 OOM 类问题的常见根源。建议重点关注 `DataProto.__getstate__()` 的重构方式和 monkeypatch 在测试中的巧妙运用。设计上的关键点是分支提前, 避免不必要的计算, 这是一个简洁有效的优化模式。未来可关注 wuxibin89 提到的 TQ 迁移, 以彻底消除序列化成本。

功能与动机

在 verl-omni 的 Issue #402 中, 用户报告在 Ascend 节点运行 wan2.2 时出现 host-RAM OOM 错误。调查发现, `DataProto.__getstate__()` 在 NumPy 序列化模式下, 会先调用 `self.batch.contiguous().consolidate()` 创建一份完整的 batch 副本, 但 NumPy 分支实际序列化的是原始 `self.batch`, 导致该副本被完全浪费。对于 Qwen-Image FlowGRPO 负载, 该单个 batch 可达约 464 MiB, 因而增大了主机内存压力。此 PR 旨在移除该无用的分配, 从根本上缓解 OOM 问题。

实现拆解

本次变更仅涉及两个文件: `verl/protocol.py` 和新增的测试文件 `tests/test_protocol_on_cpu.py`, 具体实现步骤如下:

1. 将 NumPy 分支提前: 在 `DataProto.__getstate__()` 中, 首先判断 `VERL_DATAPROTO_SERIALIZATION_METHOD` 环境变量。若是 `numpy`, 则直接返回 `serialize_tensordict(self.batch)` 结果 (若 `batch` 为 `None` 则返回 `None`), 完全跳过 `consolidate()` 调用。这样 NumPy 序列化不会产生额外的内存分配, 也不需要访问 `consolidate()` 所依赖的 `tensordict` 版本检查和空 `batch` 特殊处理。
2. 保留 torch 分支原有逻辑: 将原本的 `contiguous().consolidate()` 逻辑完整保留在默认的 `torch / torch.save()` 序列化分支中, 并保留对 `tensordict` 版本 `>=0.5.0` 的检查以及空 `batch` 的保护, 确保 `torch.cat` 错误不会在空 `TensorDict` 上触发。
3. 新增回归测试: 在 `tests/test_protocol_on_cpu.py` 中添加 `test_numpy_dataproto_serialization_skips_tensordict_consolidation`, 通过 monkeypatch 将 `TensorDict.consolidate` 替换为直接 `pytest.fail` 的函数, 并设置 `VERL_DATAPROTO_SERIALIZATION_METHOD=numpy`, 随后构造一个 `DataProto` 并执

行 `__getstate__` 和 `__setstate__`，验证 `consolidate` 绝对不会被调用，同时序列化前后的数据保持一致。

4. 配置与部署方面：无任何配置变更或部署配套改动，用户继续通过 `export VERL_DATAPROTO_SERIALIZATION_METHOD=numpy` 使用 NumPy 序列化，非 NumPy 模式行为不变。

关键文件：

- `verl/protocol.py`（模块 协议；类别 `source`；类型 `core-logic`）：核心修改文件，重排了 `DataProto` 序列化的分支逻辑，避免了 NumPy 模式下无用的 `consolidate` 调用。
- `tests/test_protocol_on_cpu.py`（模块 协议测试；类别 `test`；类型 `test-coverage`；符号 `test_numpy_dataproteo_serialization_skips_tensordict_consolidation`, `fail_on_consolidate`）：新增了针对 NumPy 序列化跳过 `consolidate` 的回归测试，通过 `monkeypatch` 确保 `consolidate` 不被调用。

关键符号： `getstate`

关键源码片段

`verl/protocol.py`

核心修改文件，重排了 `DataProto` 序列化的分支逻辑，避免了 NumPy 模式下无用的 `consolidate` 调用。

`verl/protocol.py` 中 `DataProto.__getstate__` 的修改后实现

```
def __getstate__(self):
    # 若启用 NumPy 序列化，直接序列化原始 batch，避免不必要的 consolidate。
    if os.getenv("VERL_DATAPROTO_SERIALIZATION_METHOD") == "numpy":
        # self.batch 为 None 时返回 None，保持原有行为；否则序列化自 batch。
        return (
            serialize_tensordict(self.batch) if self.batch is not None else None,
            self.non_tensor_batch,
            self.meta_info,
        )
    else:
        # torch 序列化仍调用 consolidate，以便在保存前将 TensorDict 转为连续内存。
        if version.parse(tensordict.__version__) >= version.parse("0.5.0") and self.batch is not None:
            # 空 batch 时跳过 consolidate，避免 torch.cat 错误。
            if len(self.batch.keys()) > 0:
                batch = self.batch.contiguous().consolidate()
            else:
                batch = self.batch
        else:
            batch = self.batch

    import io
    buffer = io.BytesIO()
    torch.save(batch, buffer)
```

```
buffer_bytes = buffer.getvalue()
return buffer_bytes, self.non_tensor_batch, self.meta_info
```

tests/test_protocol_on_cpu.py

新增了针对 NumPy 序列化跳过 consolidate 的回归测试，通过 monkeypatch 确保 consolidate 不被调用。

```
# tests/test_protocol_on_cpu.py 新增的回归测试

def test_numpy_dataproto_serialization_skips_tensordict_consolidation(monkeypatch):
    """NumPy serialization should not allocate an unused consolidated TensorDict."""
    # 设置环境变量为 numpy 模式。
    monkeypatch.setenv("VERL_DATAPROTO_SERIALIZATION_METHOD", "numpy")

    # 构造带张量与非张量 batch 的 DataProto。
    data = DataProto.from_dict(
        tensors={"obs": torch.arange(12).reshape(3, 4)},
        non_tensors={"labels": np.array(["a", "b", "c"], dtype=object)},
        meta_info={"step": 1},
    )

    # 若 consolidate 被调用，直接使测试失败。
    def fail_on_consolidate(*args, **kwargs):
        pytest.fail("TensorDict.consolidate() should not be called in NumPy serialization mode")

    monkeypatch.setattr(TensorDict, "consolidate", fail_on_consolidate)

    # 执行序列化与反序列化，验证数据一致。
    state = data.__getstate__()
    restored = DataProto()
    restored.__setstate__(state)

    torch.testing.assert_close(restored.batch["obs"], data.batch["obs"])
    assert restored.non_tensor_batch["labels"].tolist() == ["a", "b", "c"]
    assert restored.meta_info == {"step": 1}
```

评论区精华

Review 中有如下讨论：

- SamitHuang 询问是否需要为这种不必要的 full-batch TensorDict 分配添加回归测试（对应行的评论）。作者随后添加了上述测试，满足该要求。
- wuxibin89 询问 verl-omni 是否有计划切换到 TQ（TensorDict 的另一个序列化方案），因为 TQ 序列化成本为零。Sky-Trigger 回复说“Yes, but migrating the script to the V1 trainer will take some time.”表明这是一个中长期方向，但当前修复仍是必要的过渡步骤。
- 是否需要回归测试 (testing): 作者随后添加了回归测试，验证 consolidate 不被调用。
- 迁移到 TQ 的可能性 (design): 当前无明确计划，但指出迁移需要时间，本修复作为过渡。

风险与影响

- 风险：主要风险集中在 NumPy 序列化分支的行为变化上。由于之前 NumPy 分支虽然在逻辑上不使用 `consolidate()` 的结果，但在某些 `tensorDict` 版本下，`consolidate()` 可能会对原始 batch 产生副作用（例如将数据复制到连续内存），去除该调用后，序列化前的 batch 布局可能与之前不同。不过 `serialize_tensorDict` 本身会处理数据，故影响不大。另外，test 通过 monkeypatch 验证 `consolidate` 不被调用，但未覆盖 `self.batch` 为 `None` 的情况，不过代码逻辑中已直接返回 `None`，风险极低。
- 影响：影响范围限定在使用 NumPy 序列化方法的用户，主要缓解了主机内存（RSS）的增长。对于 Qwen-Image FlowGRPO 等以大 batch 为粒度的负载，单次序列化可节省数百 MiB 内存。对默认 torch 序列化路径无影响，功能行为保持一致。对团队而言，此修复降低了 OOM 风险，提升了大模型训练（特别是多模态或大 batch）的稳定性，并简化了 verl-omni 下游的 monkey patch 依赖。
- 风险标记：核心路径变更，测试覆盖充分

关联脉络

- PR #402 It throws some OOM error when we run wan2.2 with Verl-omni on ascend node: 此 Issue 报告了 OOM 问题，本 PR 是其上游根因修复的一部分。
- PR #423 compatibility workaround for NumPy DataProto serialization: verl-omni 的兼容性补丁，本 PR 将相同的修复直接落地到 verl，使下游可以移除该 monkey patch。