# PR #7158 完整报告

verl-project/verl

[rollout] fix: TypeError merging numpy routed_experts on partial rollout resume

合并时间：2026-07-27 17:06

原文链接：http://prhub.com.cn/verl-project/verl/pull/7158

## 执行摘要

- 一句话：修复局部 rollout 恢复时 routed_experts 类型错误
- 推荐动作：本 PR 改动简洁但关键，适合需要了解 rollout 路由机制和跨后端兼容性处理的开发人员精读。测试代码尤其值得参考，展示了如何模拟多后端差异并验证合并逻辑。

## 功能与动机

Partial rollout resume 时，FullyAsyncLLMServerClient 需要将多次 generate 返回的 routed_experts 合并。不同后端返回的数据类型不一致：vLLM 为 numpy.uint8，SGLang detokenized 模式返回只读 numpy.int32，skip_tokenizer_init 返回可写 numpy.int32。原代码使用 torch.cat 期望 torch tensor，与 numpy 不兼容导致 TypeError。需统一使用 numpy 数组操作并保证合并正确。

## 实现拆解

1. llm_server.py修改合并路由逻辑：将 torch.cat 替换为 np.concatenate，移除不再需要的 import torch，添加 import numpy as np。此改动兼容所有后端的 numpy 数组。
2. async_sglang_server.py在 skip_tokenizer_init 为 True 的分支中，将来自 scheduler tensor 的 routed_experts 显式调用 .numpy() 转换为 numpy 数组，确保与下游约定一致。
3. 新增测试文件test_llm_server_routed_experts_on_cpu.py：包含辅助函数 _routing（模拟各后端返回格式）、_markers（提取首列标识）、_install_segments（mock 多次 resume 的 generate 行为），以及两个核心测试用例 test_resume_appends_only_newly_generated_routing（验证多次 resume 时仅追加新 token 对应路由）和 test_merge_yields_numpy_and_preserves_dtype（验证合并后数组类型仍为 numpy）。

关键文件：

- verl/workers/rollout/llm_server.py（模块 生成；类别 source；类型 core-logic；符号 FullyAsyncLLMServerClient.generate）：核心修复：将合并 routed_experts 从 torch.cat 改为 np.concatenate，解决 TypeError。同时移除 torch 依赖，增加 numpy 导入。
- verl/workers/rollout/sglang_rollout/async_sglang_server.py（模块 生成；类别 source；类型 core-logic；符号 SGLangLLMServerClient.generate）：确保 skip_tokenizer_init 分支下 routed_experts 始终返回 numpy.ndarray，与 llm_server.py 的期望一致。

- tests/workers/rollout/test_llm_server_routed_experts_on_cpu.py（模块 测试；类别 test；类型 test-coverage；符号 _routing, _markers, _install_segments, fake_generate）：新增 187 行单元测试，全面覆盖 vLLM、SGLang detokenized、SGLang skip_tokenizer_init 三种后端的 routing 记录合并，确保 no regression。

关键符号：FullyAsyncLLMServerClient.generate, SGLangLLMServerClient.generate, _routing, _markers, _install_segments, fake_generate, no_wait, _client, test_resume_appends_only_newly_generated_routing, test_merge_yields_numpy_and_preserves_dtype

## 关键源码片段

### verl/workers/rollout/llm_server.py

核心修复：将合并 routed_experts 从 torch.cat 改为 np.concatenate，解决 TypeError。同时移除 torch 依赖，增加 numpy 导入。

```
# verl/workers/rollout/llm_server.py
# 位于 FullyAsyncLLMServerClient.generate 的 resume 循环中

# 之前的实现：torch.cat（要求输入为 torch.Tensor）
# 现在改为 np.concatenate，因为各后端均返回 numpy.ndarray
if output.routed_experts is not None and len(output.token_ids) > 0:
    if final_output.routed_experts is None:
        final_output.routed_experts = output.routed_experts
    else:
        # 仅将新生成的 token 对应的 routing 记录追加到已有数组
        final_output.routed_experts = np.concatenate(
            [final_output.routed_experts, output.routed_experts[-len(output.token_ids):]]
        )
```

## 评论区精华

Review 中 wuxibin89 指出应让 async_sglang_server 始终返回 np.array，与 llm_server.py 的改动对齐。该建议已被采纳，体现在第二个提交中。无其他争议或未解决问题。

- 确保 async_sglang_server 返回 numpy 数组 (design): 作者在第二个提交中实现了该要求，在 skip_tokenizer_init 分支添加了 .numpy() 调用。

## 风险与影响

- 风险：核心风险在于假设 routed_experts 始终为 numpy 数组：若未来后端返回 torch tensor 或其它类型，np.concatenate 会直接崩溃。但当前所有后端（vLLM、SGLang）均已在各自路径上确保了 numpy 输出，且测试覆盖了三种典型情况。另一风险是部分 resume 时路由切片索引 output.routed_experts[-len(output.token_ids):] 在单 token 生成时是否正确，测试用例已涵盖多 token 场景。
- 影响：影响所有使用 partial rollout 的 MoE 模型训练（vLLM 和 SGLang 后端），修复了因 TypeError 导致训练中断的问题。对不使用 partial rollout 的配置无影响。新增测试确

保回归防护。

- 风险标记：核心数据类型假设变更，测试需覆盖多后端差异

## 关联脉络

- PR #5599 [megatron] fix: Qwen3.5 LoRA & MTP support (with Megatron-Bridge): 涉及 vllm_rollout/utils.py 中 routed_experts 相关修改，与本 PR 同属 rollout 模块中的 routing 数据流改造。
- PR #7139 [sglang] fix: use _base guard in _compact_for_bucket to prevent NCCL buffer race: 同为 SGLang 后端的稳定性修复，与本 PR 关注 partial rollout 的可靠性提升方向一致。