

PR #7207 完整报告

verl-project/verl

[rollout] fix: cap partial rollout resumes at the remaining response budget

合并时间: 2026-07-31 17:37

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7207>

执行摘要

- 一句话: 修复部分 rollout 恢复时 token 超预算 bug
- 推荐动作: 值得精读。问题定位清晰, 修复最小侵入, 测试设计具有参考价值 (用 fake server 复刻公式)。关注恢复预算的扣减逻辑, 理解该逻辑有助于后续维护异步 rollout。

功能与动机

PR body 指出: 当部分 rollout 被中止并恢复时, 生成的总 token 数可能超过 `response_length`, 额外 token 生成缓慢且之后被丢弃。根因是 `agent loop` 不设置 `max_tokens`, 服务器默认公式 `min(response_length, prompt_length + response_length - len(prompt_ids))` 在恢复时错误地将已生成 token 计入 prompt 预算, 导致超发。

实现拆解

1. 根因定位: 通过 PR body 中的数学推导, 确认 `vllm_async_server.py` 的默认公式 `min(response_length, prompt_length + response_length - len(prompt_ids))` 在恢复请求上会错误地把已生成 token 计入 prompt 预算, 导致超发。
2. 修复实现: 在 `verl/workers/rollout/llm_server.py` 中新增 `_configured_response_length` 方法, 从 `actor_rollout_ref.rollout.response_length` 读取每请求预算; 在 `FullyAsyncLLMClient.generate` 中, 当 `sampling_params` 未显式设置 `max_tokens/max_new_tokens` 时, 将 `max_tokens` 置为该预算, 使恢复路径中已有的基于 `original_max_tokens` 的扣减逻辑生效。
3. 防御性拷贝: 将 `sampling_params` 复制一份, 避免修改调用方跨 turn 复用的字典, 防止副作用。
4. 测试配套: 新增 CPU 测试 `test_llm_server_response_length_cap_on_cpu.py`, 用 `FakeRolloutServer` 精确复刻服务器 `max_tokens` 公式, 验证多次恢复后总生成量不超过 `response_length`; 该测试在旧代码上失败 (生成 10104 tokens), 在新代码上通过 (8192 tokens)。

关键文件:

- `verl/workers/rollout/llm_server.py` (模块 rollout 客户端; 类别 source; 类型 core-logic; 符号 `_configured_response_length, generate`): 核心修复: 新增 `_configured_response_length` 并从配置中读取 `response_length`, 在无显式限制时写入 `max_tokens`, 激活恢复路径已有的预算扣减逻辑。

- tests/workers/rollout/test_llm_server_response_length_cap_on_cpu.py (模块 rollout 测试; 类别 test; 类型 test-coverage; 符号 FakeRolloutServer, generate, test_resumes_never_exceed_response_length, _config) : 新增 CPU 测试, 用 FakeRolloutServer 精确复刻服务器 max_tokens 公式, 验证恢复时总生成量不超过 response_length。

关键符号: _configured_response_length, generate

关键源码片段

verl/workers/rollout/llm_server.py

核心修复: 新增 _configured_response_length 并从配置中读取 response_length, 在无显式限制时写入 max_tokens, 激活恢复路径已有的预算扣减逻辑。

```
# 从配置中读取每个请求的响应 token 预算 (response_length) 。
def _configured_response_length(self) -> Optional[int]:
    rollout_config = getattr(getattr(self.config, "actor_rollout_ref", None), "rollout", None)
    response_length = getattr(rollout_config, "response_length", None)
    if isinstance(response_length, int) and response_length > 0:
        return response_length
    return None
```

```
async def generate(self, request_id, *, prompt_ids, sampling_params, **kwargs):
```

```
    # 先识别调用方是否显式设置了 token 限制
```

```
    limit_key = None
```

```
    if "max_tokens" in sampling_params:
```

```
        limit_key = "max_tokens"
```

```
    elif "max_new_tokens" in sampling_params:
```

```
        limit_key = "max_new_tokens"
```

```
    original_max_tokens = sampling_params.get(limit_key) if limit_key else None
```

```
    # 复制一份 sampling_params, 避免修改调用方复用的字典
```

```
    sampling_params = dict(sampling_params)
```

```
    if original_max_tokens is None:
```

```
        # 无显式限制时: 用配置的 response_length 固定总预算。
```

```
        # 这样恢复路径中基于 original_max_tokens 的扣减逻辑才能生效,
```

```
        # 避免恢复请求被服务器默认公式错误地授予更多 token。
```

```
        response_length = self._configured_response_length()
```

```
        if response_length is not None:
```

```
            limit_key = "max_tokens"
```

```
            original_max_tokens = response_length
```

```
            sampling_params[limit_key] = response_length
```

```
    # 后续循环中, 每次生成前都会用 original_max_tokens 减去已生成 token 数
```

评论区精华

Review 中仅 HollowMan6 给出了 APPROVED (LGTM), 无其它实质讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
 - `_configured_response_length` 依赖配置中存在 `actor_rollout_ref.rollout.response_length`，若配置缺失或非正整数，则回退到旧行为；但正常训练配置必然包含该项。
 - 修复改变了 `sampling_params` 的副本，不直接影响调用方，但后续代码若依赖原始对象身份可能受影响。
 - 测试仅覆盖 CPU，未进行 GPU 端到端验证，但 bug 本质是整数算术，CPU 测试精确复刻了服务器公式。
- 影响：
 - 影响所有使用 `FullyAsyncLLMServerClient` 的路径，包括 `agent loop` 和异步训练（`trainer_colocate_async.py`、`trainer_separate_async.py`）。
 - 修复后，恢复请求的每步预算限制正确，总生成 token 不超过 `response_length`，避免无效解码开销，提升整体效率。
 - 对正常首次请求无影响，因为首次请求时 `len(prompt_ids)` 为真实 prompt 长度，公式结果与配置一致。
 - 风险标记：缺少 GPU 端到端验证，配置缺失时回退旧行为，`sampling_params` 拷贝行为变化

关联脉络

- PR #7158 [rollout] fix: TypeError merging numpy routed_experts on partial rollout resume: 同一 partial rollout resume 路径，且修改了同目录的测试文件 `test_llm_server_routed_experts_on_cpu.py`，本 PR 也复跑了该测试。
- PR #7204 [rollout] fix: decode per-turn LLM tokens in traces: 同属 `agent_loop` 回放路径，涉及 `agent_loop.py` 与 `rollout_trace`，与本 PR 的 `agent_loop` 预算问题同源。