PR #7565 完整报告

verl-project/verl

[rollout] fix: surface vLLM prefix-cache hit counts in TokenOutput

合并时间：2026-08-28 10:08

原文链接：http://prhub.com.cn/verl-project/verl/pull/7565

执行摘要

- 一句话：在 TokenOutput 中暴露 vLLM prefix-cache 命中统计
- 推荐动作：该 PR 值得关注，尤其对于使用 vLLM prefix-cache 或需要缓存命中统计的团队。它展示了如何在框架中暴露底层引擎的统计信息，并处理了异步恢复场景的数据一致性。建议结合实际场景评估是否需要在默认 agent loop 中透出该指标。

功能与动机

根据 PR 描述，prefix-cache 命中计数当前对 TokenOutput 消费者不可见，因为 vLLM 异步服务器未暴露 per-request 的 num_cached_tokens，且 FullyAsyncLLMServerClient.generate 在 partial-rollout 恢复时重建了 TokenOutput，导致即使修复第一点，完全异步模式也永远报告 0 次缓存命中。作者需要在自定义 agent loop 中获取该统计，但没有其他途径。

实现拆解

本 PR 的改动分为两个层面：

1. 在 vLLM 异步服务器暴露 num_cached_tokens：在 verl/workers/rollout/vllm_rollout/vllm_async_server.py 的 generate 方法中，从 final_res 对象读取 num_cached_tokens 并写入 extra_fields。这使同步模式（直接返回服务器 TokenOutput）能正确暴露缓存命中数。
2. 在 FullyAsyncLLMServerClient 中跨恢复迭代携带统计：在 verl/workers/rollout/llm_server.py 的 generate 方法中，新增 num_cached_tokens 变量，在循环中捕获首次 prefill 的命中计数，并在最终输出时写入 final_output.extra_fields['num_cached_tokens']。这确保 partial-rollout 恢复时不会丢失初始 prefill 的缓存命中信息。 测试方面，PR 仅进行了管道改动，未新增单元测试，但作者声明已在 agent-loop 训练中验证缓存命中统计从恒为 0 变为预期值。

关键文件：

- verl/workers/rollout/vllm_rollout/vllm_async_server.py（模块 vLLM 服务器；类别 source；类型 core-logic；符号 generate）：这是暴露 num_cached_tokens 的源头，从 vLLM RequestOutput 中读取字段并写入 extra_fields，是同步模式可见性的基础。
- verl/workers/rollout/llm_server.py（模块 异步服务器；类别 source；类型 core-logic；符号 FullyAsyncLLMServerClient.generate）：这是关键的可观测性修复，确保完全异步模式下跨恢复迭代保留缓存命中计数，否则消费者始终得到 0。

关键符号：generate

关键源码片段

verl/workers/rollout/llm_server.py

这是关键的可观测性修复，确保完全异步模式下跨恢复迭代保留缓存命中计数，否则消费者始终得到 0。

```
# verl/workers/rollout/llm_server.py
# 在 generate 循环开始前初始化：
num_cached_tokens = None

# 在循环内，每次迭代后捕获首次 prefill 的命中数：
if num_cached_tokens is None:
    num_cached_tokens = output.extra_fields.get("num_cached_tokens")

# 在循环结束后，写入最终 TokenOutput：
final_output.extra_fields["num_cached_tokens"] = num_cached_tokens
```

评论区精华

Review 中仅有一条讨论线，由 wuxibin89 提出疑问：

> Where is num_cached_tokens consumed?

作者 emmericp 回应：

> Not in the default agent loops, I'm using it in a custom agent loop for custom statistics. Without this patch there is no good way to get that information at all.

讨论表明该字段目前仅在自定义 agent loop 中被消费，不属于默认路径。评审者随后批准了该 PR，说明疑虑已解决。

- num_cached_tokens 的使用位置 (question): 作者回复仅用于自定义 agent loop，不在默认路径。

风险与影响

- 风险：风险较低，但需注意：
  1. 兼容性风险：num_cached_tokens 从 final_res 对象动态读取，使用 getattr 默认值为 None，对 vLLM 版本兼容性较好，但如果 vLLM 对象缺少该属性，则消费者会得到 None，需确保消费者能处理 None。
  2. 语义一致性：在 FullyAsyncLLMServerClient 中，只携带首次 prefill 的命中计数，可能无法反映多次恢复的累计缓存命中，但这是有意设计，与单次 prefill 语义一致。
  3. 无测试覆盖：PR 未新增单元测试，对于统计上报逻辑，建议后续补充测试以保障回归。
- 影响：影响范围较窄：
  - 用户 / 系统：自定义 agent loop 或 OpenAI 兼容端点的用户可获取 prefix-cache 命中统计，用于监控或优化缓存策略。默认 agent loop 不受影响。

- 团队：增强了 vLLM 集成层的可观测性，为性能分析和缓存调优提供数据基础。
- 影响程度：低。改动纯增量，无 breaking change。
- 风险标记：缺少测试覆盖

关联脉络

- PR #7508 [vllm] fix: honor explicit False on Optional[bool] engine args in CLI serialization: 同样涉及 vLLM rollout 参数和 TokenOutput 的序列化处理，属于同一模块的改进。
- PR #7539 [ray] fix: skip unused TensorDict consolidation in NumPy DataProto serialization: 同样优化了 rollout 数据传递路径，与本 PR 相关。