# PR #6270 完整报告

verl-project/verl

[worker] feat: support log memory in engine worker

合并时间：2026-05-12 15:16

原文链接：http://prhub.com.cn/verl-project/verl/pull/6270

## 执行摘要

- 一句话：engine worker 添加内存日志支持
- 推荐动作：该 PR 变更较小且逻辑清晰，但其 review 讨论揭示了分布式内存监控中的常见陷阱。建议维护者根据 review 意见进一步优化：将 CPU 内存改为进程级测量（使用 psutil.Process().memory_info().rss），并考虑将内存指标置于 all-gather 之前以确保分布式一致性。

## 功能与动机

PR body 明确说明"support log memory in engine worker"，旨在为 engine worker 添加内存使用日志，便于性能监控和调试。

## 实现拆解

1. 新增导入依赖：在 verl/workers/engine_workers.py 中增加 import psutil 和 from verl.utils.device import get_torch_device，以获取进程级内存信息。
2. 移除冗余 TODO：删除 _postprocess_output 方法中已被注释掉的三行内存测量占位代码（原先为 TODO 标记）。
3. 添加内存指标：在 _postprocess_output 方法的末尾，执行完 all-gather 后，新增三个 metric 键值对：
   - perf/max_memory_allocated_gb：GPU 当前分配的显存（GB）。
   - perf/max_memory_reserved_gb：GPU 当前预留的显存（GB）。
   - perf/cpu_memory_used_gb：系统 CPU 总内存使用量（GB）。
4. 指标收集时机：在分布式 all-gather 之后添加，因此这些指标仅反映单个 rank 的本地值，而非全局聚合值。

关键文件：

- verl/workers/engine_workers.py（模块 引擎工作器；类别 source；类型 dependency-wiring；符号 _postprocess_output）：核心变更文件，在 _postprocess_output 中添加了三个内存指标，并新增了 psutil 和 get_torch_device 的导入。

关键符号：_postprocess_output

## 关键源码片段

### verl/workers/engine_workers.py

核心变更文件，在 _postprocess_output 中添加了三个内存指标，并新增了 psutil 和 get_torch_device 的导入。

```
# verl/workers/engine_workers.py (head)
# 在 _postprocess_output 方法末尾添加内存指标记录
# 注意：这些 metric 放在 all-gather 之后，因此只反映本地 rank 的值

def _postprocess_output(self, output, *, global_token_num, delta_time, forward_only, images_
seqlens):
    # ... 原有的 loss、grad_norm、lr 处理 ...

    if dp_group is not None:
        final_metrics = allgather_dict_into_dict(data=metrics, group=dp_group)
    else:
        final_metrics = metrics

    if lr is not None:
        final_metrics["lr"] = lr

    # 新增：记录 GPU 显存分配量（当前进程分配的显存）
    final_metrics["perf/max_memory_allocated_gb"] = get_torch_device().max_memory_allocated()
    / (1024**3)
    # 新增：记录 GPU 显存预留量（GPU 驱动预留的显存）
    final_metrics["perf/max_memory_reserved_gb"] = get_torch_device().max_memory_reserved() /
    (1024**3)
    # 新增：记录系统 CPU 内存使用量（⚠🤔 这是系统级，非进程级）
    final_metrics["perf/cpu_memory_used_gb"] = psutil.virtual_memory().used / (1024**3)

    # TODO: confirm the mtp loss IS same across dp
    for k, v in final_metrics.items():
        if k.startswith("mtp_losses"):
            # ...
```

## 评论区精华

review 中 gemini-code-assist[bot] 提出了两个重点关注：

1. CPU 内存指标误导性：psutil.virtual_memory().used 返回的是整个宿主系统的内存使用量，在多 worker 场景下每个 worker 会报告相同的聚合值，无法区分单个 worker 的占用或定位内存泄漏，建议改为进程级内存测量。
2. 指标放置位置问题：三个内存指标被放在 allgather_dict_into_dict 调用之后，这意味着最终输出的指标只反映单个 rank 的本地值，而不是经过 all-gather 后的聚合值，对于分布式场景不够准确。此外函数内多次调用 get_torch_device() 存在冗余。

wucong25 审核并两次 APPROVED，未留下评论。

- CPU 内存指标应为进程级而非系统级 (correctness): 未在 PR 中修复，但 review 已指出此问题，后续可跟进优化。

## 风险与影响

- 风险：
  1. 指标准确性风险：CPU 内存使用量使用系统级 psutil.virtual_memory().used，在多 worker 同机部署时每个 worker 报告相同值，可能误导运维决策。
  2. 分布式一致性风险：指标在 all-gather 后添加，不会参与分布式归约，导致最终输出的内存指标仅来自单个 rank，在多节点场景下可能不具代表性。
  3. 依赖新增风险：引入 psutil 库，如果环境未安装可能导致运行时 ImportError；但该库为常见系统监控库，风险较低。
- 影响：对用户和系统的影响较小：
  - 用户影响：训练日志中会新增三个内存相关的 metric，有助于性能分析和诊断。
  - 系统影响：仅新增三次函数调用和一次库导入，对性能无实质性影响。
  - 团队影响：提供更丰富的监控指标，但指标定义需要后续澄清（如 CPU 内存是否应为进程级）。
  - 风险标记：准确性顾虑，缺少测试覆盖

## 关联脉络

- 暂无明显关联 PR