PR #5760 完整报告

verl-project/verl

[perf] feat: support partial-token window profiling for rollout in verl

合并时间：2026-06-05 10:34

原文链接：http://prhub.com.cn/verl-project/verl/pull/5760

执行摘要

- 一句话：新增 rollout 部分 token profiling 窗口配置
- 推荐动作：该 PR 值得精读，特别是 config.py 中字段添加与验证的方式，以及 build_vllm_profiler_args 中的映射逻辑。设计上采用了半开区间，与后端惯用参数一致。代码组织上将 TorchMemoryProfiler 独立成模块，体现了良好的模块化思想。建议在后续重构中提取公共验证函数。

功能与动机

在长上下文强化学习中，对完整 *decode* 进行 profiling 会产生非常大的 *trace* 文件。仅收集关键的 token 区间（key token range）可以减少数据体积并加快解析分析速度。verl 需要暴露 profile_token_start / profile_token_end 参数来支持窗口式 profiling，并映射到 vLLM 和 SGLang 后端。

实现拆解

1. 配置字段定义：在 TorchProfilerToolConfig 和 NPUToolConfig 中添加 profile_token_start、profile_token_end 可选 int 字段，并在 __post_init__ 中增加验证逻辑，确保 start < end 且非负。
2. 后端参数映射：在 build_vllm_profiler_args 中将 profile_token_start 映射为 delay_iterations，将 end-start 映射为 max_iterations；在 build_sglang_profiler_args 中映射为 start_step 和 num_steps。
3. 代码抽取：将原先内联在 profile.py 的 TorchMemoryProfiler 类移至独立的 torch_memory_profile.py 文件，并更新 profile.py 中的导入为延迟导入，同时清理不再需要的 memory_utils 导入。
4. 配置同步：更新 rollout.yaml 和 profiler.yaml 模板，加入新字段的默认值 null；同步更新所有 _generated_*.yaml 以保持配置一致性。
5. 测试与文档：在 test_server_profiler.py 中新增 4 个测试用例覆盖 vLLM、SGLang 以及 NPU 配置下的窗口映射。更新 ascend_profiling_en.rst 和 torch_profiling.md 文档说明新参数用法。

关键文件：

- verl/utils/profiler/torch_memory_profile.py（模块 内存分析；类别 source；类型 dependency-wiring；符号 TorchMemoryProfiler, init, start, stop）：新文件，将

TorchMemoryProfiler 从 profile.py 拆分出来，与其他 profiler 工具文件对齐，便于维护。

- verl/utils/profiler/profile.py（模块 分析器；类别 source；类型 dependency-wiring；符号 DistProfiler, start, stop）：核心调度器，移除了 TorchMemoryProfiler 类并改为延迟导入，同时添加了方法文档注释和少量逻辑修正。
- verl/utils/profiler/config.py（模块 配置层；类别 source；类型 core-logic；符号 TorchProfilerToolConfig, NPUToolConfig, build_vllm_profiler_args, build_sglang_profiler_args）：核心配置变更：在 TorchProfilerToolConfig 和 NPUToolConfig 中添加窗口字段及验证，在 build_vllm_profiler_args 和 build_sglang_profiler_args 中实现参数映射。
- tests/utils/test_server_profiler.py（模块 测试；类别 test；类型 test-coverage；符号 test_build_vllm_profiler_args_with_profile_window, test_build_vllm_profiler_args_with_npu_profile_window, test_build_sglang_profiler_args_with_profile_window）：新增 4 个测试用例，验证 profile_token_start/end 在 vLLM、SGLang、NPU 下的映射结果。
- verl/trainer/config/rollout/rollout.yaml（模块 配置；类别 config；类型 configuration）：配置文件模板，为 npu 和 torch profiler 添加 profile_token_start/end 默认值 null。
- verl/trainer/config/profiler/profiler.yaml（模块 配置；类别 config；类型 configuration）：profiler 独立配置模板，同步添加 profile_token_start/end 默认 null。
- docs/ascend_tutorial/dev_guide/performance/ascend_profiling_en.rst（模块 文档；类别 docs；类型 documentation）：更新 Ascend profiling 文档，说明新参数用法。
- docs/perf/torch_profiling.md（模块 文档；类别 docs；类型 documentation）：更新 torch profiling 文档，说明新参数用法。

关键符号：build_vllm_profiler_args, build_sglang_profiler_args, TorchMemoryProfiler.start, TorchMemoryProfiler.stop, test_build_vllm_profiler_args_with_profile_window, test_build_sglang_profiler_args_with_profile_window

关键源码片段

verl/utils/profiler/config.py

核心配置变更：在 TorchProfilerToolConfig 和 NPUToolConfig 中添加窗口字段及验证，在 build_vllm_profiler_args 和 build_sglang_profiler_args 中实现参数映射。

```
@dataclass
class TorchProfilerToolConfig(BaseConfig):
    """Torch profiler tool config."""
    contents: list[str] = field(default_factory=list)
    discrete: bool = False
    # Start collecting profiler data from this response-token index.
    # None means collect from the beginning.
    profile_token_start: Optional[int] = None
    # Stop collecting profiler data at this response-token index (exclusive).
    # None means collect until the end.
    profile_token_end: Optional[int] = None
```

```
    name: str = "torch"

    def __post_init__(self) -> None:
        __support_contents = ["cuda", "cpu", "memory", "shapes", "stack"]
        for content in self.contents:
            assert content in __support_contents, (
                f"Profiler contents only supports {__support_contents}, but gets {content}"
            )
        assert isinstance(self.contents, list), \
            f"Profiler contents must be of type list, got {type(self.contents)}"
        start = self.profile_token_start
        stop = self.profile_token_end
        for name, value in (("profile_token_start", start), ("profile_token_end", stop)):
            if value is not None:
                assert isinstance(value, int), f"{name} must be int or None, got {type(value)}"
                assert value >= 0, f"{name} must be >= 0, got {value}"
        if start is not None and stop is not None:
            assert stop > start, f"profile_token_end must be > profile_token_start, got start={start},
            stop={stop}"


def build_vllm_profiler_args(profiler_config, tool_config, rank):
    # ... existing code ...
    profile_token_start = getattr(tool_config, "profile_token_start", None)
    profile_token_end = getattr(tool_config, "profile_token_end", None)

    # vLLM uses 0 to indicate immediate start / no upper bound.
    delay_iterations = profile_token_start if profile_token_start is not None else 0
    max_iterations = (profile_token_end - profile_token_start) \
                     if (profile_token_start is not None and profile_token_end is not None) else 0
    # ... continue building args ...
```

评论区精华

代码审查中，gemini-code-assist[bot] 指出 TorchProfilerToolConfig 与 NPUToolConfig 中的验证逻辑完全重复，建议提取为模块级辅助函数 _validate_profiling_window(start, stop) 以遵循 DRY 原则。tardis-key 询问在 fullyasync 模式下的测试情况，mengchengTang 回复已验证通过，fullyasync 调用的是 replica 的 profiler 接口，无需额外改动即可支持。

- 验证逻辑重复 (design): PR 作者未回复此建议，目前验证逻辑仍有两份拷贝。建议后续迭代中提取公共函数。
- fullyasync 兼容性 (testing): mengchengTang 回复已验证，fullyasync 调用的是 replica 的 profiler 接口，无需额外改动即可支持。

风险与影响

- 风险：

1. 配置兼容性风险：新字段默认为 None，不会影响已有配置，但用户如果误设置导致 start >= end 或负数，验证会报错，属预期行为。
2. 验证逻辑代码重复：两个 Config 类的验证逻辑完全一致，后续若修改窗口语义（如改为闭区间）需要同步两处，有遗漏风险。建议按审查意见提取公共函数。
3. 后端映射语义需同步：vLLM 和 SGLang 的 delay_iterations/max_iterations 语义可能随版本变化，需要保持同步。
4. 模块抽取影响外部引用：TorchMemoryProfiler 从 profile.py 移出，但 profile.py 中已调整为延迟导入，且该符号被重新导出，应保持兼容。 - 影响：用户：现在可以通过 actor_rollout_ref.rollout.profiler.tool_config.npu.profile_token_start=20 actor_rollout_ref.rollout.profiler.tool_config.npu.profile_token_end=80 精确控制 profiling 收集窗口，减少 trace 体积。系统：无性能影响，功能仅在启用 profiler 时生效。团队：需要维护两套验证逻辑，建议后续重构。 - 风险标记：验证逻辑代码重复，后端映射语义需同步，配置兼容性依赖默认值

关联脉络

- 暂无明显关联 PR