

# PR #56016 完整报告

vllm-project/vllm

[CPU][Profiler] Group torch profiler tables by input shape when record\_shapes is on

合并时间: 2026-09-14 17:16

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56016>

## 执行摘要

- 一句话: CPU profiler 表格按输入形状分组
- 推荐动作: 该 PR 改动小而清晰, 值得快速浏览以了解 profiler 配置和平台条件分支的写法。若你是 CPU 性能分析相关开发者, 可以关注其手动验证方式, 并考虑未来为该分支补充单元测试。

## 功能与动机

PR body 中指出: 不分组时 CPU dump 会把同一 op 的所有调用折叠为一行, 掩盖 prefill vs decode 以及 GEMM 尺寸差异, 导致性能分析难以定位瓶颈。作者通过在 `_build_profiler_table` 中为 `key_averages` 传入 `group_by_input_shape` 解决该问题, 并明确该行为仅在 `current_platform.is_cpu()` 且 `torch_profiler_record_shapes` 为真时生效, GPU 不受影响。

## 实现拆解

1. 新增平台导入: 在 `vllm/profiler/wrapper.py` 顶部加入 `from vllm.platforms import current_platform`, 用于在构建表格时判断当前运行平台。
2. 改造 `_build_profiler_table` 方法: 将原来的 `self.profiler.key_averages()` 改为先计算 `group_by_input_shape` 布尔值 (`current_platform.is_cpu()` and `self.profiler_config.torch_profiler_record_shapes`), 然后调用 `self.profiler.key_averages(group_by_input_shape=group_by_input_shape)` 得到 `averages` 对象, 再交给 `table()` 渲染。这样 CPU + `record_shapes` 开启时表格按输入形状拆分, 否则与原来行为一致。
3. 无配套测试: 本次变更未新增或修改测试文件。PR body 提供了手动验证步骤 (CPU throughput bench) 和预期输出样例, 但仓库内没有自动测试覆盖该分支。

关键文件:

- `vllm/profiler/wrapper.py` (模块 分析器; 类别 source; 类型 dependency-wiring; 符号 `_build_profiler_table`): 唯一变更文件, 集中实现了 CPU profiler 表格按输入形状分组的功能, 涉及平台判断和 `key_averages` 参数传递。

关键符号: `_build_profiler_table`

## 关键源码片段

## vllm/profiler/wrapper.py

唯一变更文件，集中实现了 CPU profiler 表格按输入形状分组的功能，涉及平台判断和 key\_averages 参数传递。

```
# vllm/profiler/wrapper.py
# 新增平台导入，用于判断当前运行环境
from vllm.platforms import current_platform

def _build_profiler_table(
    self,
    sort_key: str,
    row_limit: int | None = None,
) -> str:
    # 仅在 CPU 平台且用户开启 shapes 记录时按输入形状分组，
    # GPU 平台保持原有聚合行为，避免改变现有输出
    group_by_input_shape = (
        current_platform.is_cpu()
        and self.profiler_config.torch_profiler_record_shapes
    )
    averages = self.profiler.key_averages(group_by_input_shape=group_by_input_shape)
    if row_limit is None: # 使用 profiler 默认行数限制 100
        return averages.table(sort_by=sort_key)
    return averages.table(
        sort_by=sort_key,
        row_limit=row_limit,
    )
```

## 评论区精华

review 中无实质技术讨论，主要流程如下：

- claude[bot] 指出该 PR 来自 fork，自动 review 被禁用，需要维护者手动触发。
- mergify[bot] 报告 pre-commit 检查失败，作者随后修复并致歉。
- 维护者 bigPYJ1151 和 reviewer amukho 均批准，bigPYJ1151 提示修复 pre-commit 后即 LGTM。
- pre-commit 检查失败与修复 (style): 作者修复后，维护者 bigPYJ1151 与 amukho 均批准通过。

## 风险与影响

- 风险：
  - 变更范围极小，仅影响 vllm/profiler/wrapper.py 中的 \_build\_profiler\_table，不涉及生产推理路径。
  - 风险点在于缺少自动测试：若 CPU 平台未来变更 current\_platform 语义或 torch 的 key\_averages 接口行为变化，该条件分支可能回归且 CI 无法感知。

- `group_by_input_shape` 仅当 `record_shapes` 开启时生效，未开启时行为与旧版完全一致，兼容性风险低。
- 影响：
  - 对用户：CPU 平台上使用 `torch profiler` 且开启 `torch_profiler_record_shapes` 时，生成的 `profiler_out_{rank}.txt` 表格会更细粒度，便于区分 `prefill/decode` 和不同 GEMM 形状的耗时；GPU 用户无感知，默认行为不变。
  - 对系统：不改变运行时性能，仅影响 `profiler` 输出格式。
  - 对团队：为 CPU 性能分析工作流提供了更好的可观测性，后续性能优化可基于更准确的表格定位热点。
  - 风险标记：缺少测试覆盖，条件分支仅在 CPU 下生效

## 关联脉络

- 暂无明显关联 PR