

PR #47725 完整报告

vllm-project/vllm

[BugFix][LoRA] Refresh punica metadata when LoRA slots are reassigned under an unchanged mapping

合并时间: 2026-07-07 23:53

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47725>

执行摘要

- 一句话: 修复 LoRA 槽位重分配导致 punica 元数据过期
- 推荐动作: 值得精读的设计决策: 通过额外字段扩展现有 memo, 而非完全移除优化。该模式可在类似缓存失效场景复用。

功能与动机

FIX #47724。运行时 `add_lora()` 可能 LRU 驱逐正在解码的适配器并旋转 `lora_index_to_id`, 但 in-flight 请求集的 `LoRAMapping` 未变, memo 命中跳过 punica 刷新, 导致 `token_lora_indices` 仍编码旧槽位, 请求实际使用另一个适配器的权重。详见 Issue 附带的完整分析及自助复现脚本。

实现拆解

1. 新增 `_last_slot_layout` 字段 (`vllm/lora/model_manager.py`, `__init__` 中初始化)。
2. 修改 `set_adapter_mapping`: 在比较 `_last_mapping` 的基础上, 增加 `_last_slot_layout ! = slot_layout` 条件 (`slot_layout` 由 `tuple(self.lora_index_to_id)` 即时获得); 任一变时即执行 `_set_adapter_mapping` 并更新两个 memo。
3. 修复后, 每步无变更时 (常见情况) 仍命中 memo, 保留性能。
4. 新增回归测试 (`tests/lora/test_lora_manager.py`, `test_set_adapter_mapping_refreshes_after_slot_reassignment`): 在 manager 层面驱动精确生产序列——两个适配器解码, 外部激活 LRU 驱逐后重分配槽位, 验证 `token_lora_indices` 反映新布局。

关键文件:

- `vllm/lora/model_manager.py` (模块 LoRA 管理; 类别 source; 类型 data-contract; 符号 `set_adapter_mapping`, `LoRAModelManager.init`): 核心修复: 新增 `_last_slot_layout` 字段, 修改 `set_adapter_mapping` 使槽位布局变化时也能触发 punica 元数据刷新。
- `tests/lora/test_lora_manager.py` (模块 LoRA 管理; 类别 test; 类型 test-coverage; 符号 `test_set_adapter_mapping_refreshes_after_slot_reassignment`): 新增回归测试 `test_set_adapter_mapping_refreshes_after_slot_reassignment`, 精确复现槽位重分配后元数据刷新的场景。

关键符号: `LoRAModelManager.set_adapter_mapping`, `LoRAModelManager.init`

关键源码片段

vllm/lora/model_manager.py

核心修复：新增 `_last_slot_layout` 字段，修改 `set_adapter_mapping` 使槽位布局变化时也能触发 punica 元数据刷新。

```
# vllm/lora/model_manager.py (head 版本)
```

```
def __init__(self, model, ...):
    # ... 原有初始化 ...
    self._last_mapping: LoRAMapping | None = None
    self._last_slot_layout: tuple[int | None, ...] | None = None # 新增：记录上一次的槽位布局
    # ...

def set_adapter_mapping(self, mapping: LoRAMapping) -> None:
    # The punica metadata derives from the slot layout as well as the
    # mapping: an out-of-band add_lora() can LRU-evict and reassign slots
    # while the running batch, and thus the mapping, is unchanged.
    slot_layout = tuple(self.lora_index_to_id) # 从当前槽位映射构建快照
    if self._last_mapping != mapping or self._last_slot_layout != slot_layout:
        self._set_adapter_mapping(mapping)
        self._last_mapping = mapping
        self._last_slot_layout = slot_layout
    # 若 mapping 和 slot_layout 均未变，则跳过刷新，保持原有性能
```

tests/lora/test_lora_manager.py

新增回归测试 `test_set_adapter_mapping_refreshes_after_slot_reassignment`，精确复现槽位重分配后元数据刷新的场景。

```
# tests/lora/test_lora_manager.py (head 版本)
```

```
@pytest.mark.parametrize("device", DEVICES)
def test_set_adapter_mapping_refreshes_after_slot_reassignment(
    default_vllm_config, dist_init, dummy_model, device
):
    # 场景：运行时 add_lora() 导致 LRU 驱逐并重分配槽位，同时运行的批次
    # 及其中 LoRAMapping 保持不变，此时 punica 元数据必须重新派生。
    # ... ( 创建 manager 和三个 LoRA 适配器，省略模型创建细节 )
    manager = LRUCacheLoRAModelManager(
        model, 2, 2, 2,
        LoRAConfig(max_lora_rank=8, max_cpu_loras=3, max_loras=2, lora_dtype=DEFAULT_
        DTYPE),
        device=device,
        vllm_config=default_vllm_config,
    )
    punica_wrapper = manager.punica_wrapper_mapping[DEFAULT_LANGUAGE_WRAPPER_KEY]

    # 先激活两个适配器，占据 slot 0 和 slot 1
    assert manager.add_adapter(model_lora1)
```

```

assert manager.activate_adapter(1)
assert manager.add_adapter(model_lora2)
assert manager.activate_adapter(2)
assert manager.lora_index_to_id == [1, 2]

# 模拟两个 in-flight 请求各一个 token，分别使用适配器 1 和 2
manager.set_adapter_mapping(LoRAMapping((1, 2), (1, 2)))
assert punica_wrapper.token_lora_indices.tolist() == [0, 1] # slot 0-1 对应 adapter 1-2

# out-of-band add_lora(): 激活适配器 3 会 LRU 驱逐适配器 1（被驱逐的槽位），
# 然后再激活 1 和 2，它们落入了交换后的槽位，但批次不变
assert manager.add_adapter(model_lora3)
assert manager.activate_adapter(3) # 驱逐 slot 0 中的 adapter 1
assert manager.activate_adapter(1) # 适配器 1 重新激活，可能占用 slot 0（若被驱逐后空闲）
assert manager.activate_adapter(2) # 适配器 2 重新激活
assert manager.lora_index_to_id == [2, 1] # 槽位已互换

# 相同的 LoRAMapping，但 token_lora_indices 必须反映新槽位布局
manager.set_adapter_mapping(LoRAMapping((1, 2), (1, 2)))
assert punica_wrapper.token_lora_indices.tolist() == [1, 0] # 现在 slot 0 是 adapter 2，slot 1
是 adapter 1

```

评论区精华

无人工 review 讨论。仅 bot 评论和 maintainer 简单 approval。

- 暂无高价值评论线程

风险与影响

- 风险：本修复非常聚焦：仅修改 `set_adapter_mapping` 中的 `memo` 条件，新增一个字段 `_last_slot_layout`。风险极低，回归风险局限于 LoRA 槽位变更场景。已有端到端复现脚本和单元测试验证。
- 影响：用户影响：修复生产环境中运行时添加 LoRA 适配器可能导致的静默错误解码，直接影响正确性。系统影响：无性能退化（无变更时仍命中 `memo`）。团队影响：极小的核心路径改动，易于审查。
- 风险标记：核心路径变更，测试覆盖充分

关联脉络

- PR #36187 LoRA shrink kernel contiguity: PR body 提到搜索过相关问题，该 PR 是无关项
- PR #39935 sleep/wake with LoRA: PR body 提到搜索过相关问题，该 PR 是无关项