

PR #45679 完整报告

vllm-project/vllm

[Test][KV Connector] Add request_finished fence population tests for offloading scheduler

合并时间: 2026-06-18 13:13

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45679>

执行摘要

- 一句话: 为 KV 卸载调度器添加 request_finished fence 回归测试
- 推荐动作: 此 PR 的测试设计模式值得学习, 尤其是通过 post_step_fn 回调捕获中间状态, 避免直接调用内部方法。所有测试保持一致的 runner.run() 入口, 便于维护。

功能与动机

issue #42761 报告了 kv_cache_offload 因 `_block_id_to_pending_jobs[bid]` 的 KeyError 崩溃。该 bug 在 #42959 中修复, 但缺少针对 fence 填充与冲刷的回归测试。本 PR 新增了覆盖 fence 索引准确性的测试, 防止类似回归。

实现拆解

1. 添加 post_step_fn 回调: 在 tests/.../utils.py 的 _run() 方法中增加 post_step_fn 参数, 在每次 update_from_output() 后调用, 允许测试捕获中间状态。
2. 增强现有 fence 测试: 修改 test_fence_at_update_state_after_alloc 和 test_fence_at_build_store_jobs, 通过 post_step_fn 捕获 _block_id_to_pending_jobs 的快照, 并断言 fence 被正确填充。
3. 新增三个测试:
 - test_request_finished_with_pending_stores_populates_fence: 验证请求完成时 fence 记录非滑动窗口块 ID, 并在块重用后清空。
 - test_multiple_in_flight_stores_all_flushed_by_fence: 模拟多个在途 store 作业, 验证它们通过 fence 在块重用时刻一起冲刷。
 - test_request_finished_mixed_full_attn_and_sliding_window: 使用混合注意力架构, 确保滑动窗口块不进入 fence 且最终清理干净。所有测试均通过 runner.run() 入口执行, 保持一致性。

关键文件:

- tests/v1/kv_connector/unit/offloading_connector/test_scheduler.py (模块 卸载调度; 类别 test; 类型 test-coverage; 符号 capture_fence, test_request_finished_with_pending_stores_populates_fence, test_multiple_in_flight_stores_all_flushed_by_fence, test_request_finished_mixed_full_attn_and_sliding_window) : 包含所有新增和增强的测试, 是 PR 的核心。

- tests/v1/kv_connector/unit/offloading_connector/utils.py (模块 测试辅助; 类别 test; 类型 test-coverage; 符号 _run) : 提供 post_step_fn 回调支持, 使测试能捕获中间状态。

关键符号: capture_fence, test_request_finished_with_pending_stores_populates_fence, test_multiple_in_flight_stores_all_flushed_by_fence, test_request_finished_mixed_full_attn_and_sliding_window, _run

关键源码片段

tests/v1/kv_connector/unit/offloading_connector/test_scheduler.py

包含所有新增和增强的测试, 是 PR 的核心。

```
# 定义 capture_fence 闭包, 用于在每步更新后捕获 fence 快照
fence_snapshots: list[dict] = []

def capture_fence():
    fence_snapshots.append(
        dict(runner.connector_scheduler._block_id_to_pending_jobs)
    )

# 通过 post_step_fn 传入 runner.run, 在每一步 update_from_output 后调用
runner.run(
    decoded_tokens=[EOS_TOKEN_ID],
    complete_transfers=False,
    post_step_fn=capture_fence,
)

# 验证 fence 在某个步骤被填充了非滑动窗口块的 ID
assert runner.connector_scheduler._block_id_to_pending_jobs
populated_fence = next((f for f in fence_snapshots if f), None)
assert populated_fence is not None, "Fence was never populated"
assert len(populated_fence) > 0, "Fence is empty"
```

tests/v1/kv_connector/unit/offloading_connector/utils.py

提供 post_step_fn 回调支持, 使测试能捕获中间状态。

```
def _run(
    self,
    decoded_tokens: list[int],
    complete_transfers: bool,
    post_step_fn: Callable[[], None] | None = None,
):
    """
    Runs multiple engine (scheduler + worker) steps.
    Assumes a single request is running.

    Args:
        decoded_tokens: the tokens to yield at each step.
        complete_transfers: complete transfers immediately
```

```

    post_step_fn: optional callback invoked after each step's
        update_from_output(), before the next schedule().
"""
# ...
while True:
    # ...
    self.scheduler.update_from_output(scheduler_output, model_runner_output)

    # 如果提供了 post_step_fn, 则在每次 update 后调用
    if post_step_fn is not None:
        post_step_fn()
    # ...

```

评论区精华

Review 中 orozery 提出三点建议:

1. 使用 `runner.run()` 替代直接调用 `schedule()` 以保持一致性。
 2. 移除冗余的 `side_effect` 重置。
 3. 确保 `mixed` 测试验证最终 `fence` 清空。作者通过添加 `post_step_fn` 回调并调整测试实现全部采纳。#45595 合并后, 作者进一步适配非阻塞 `drain`, 将断言从 `all-finished flush` 改为 `block-reuse fence flush`。
- 使用 `runner.run` 统一测试入口 (design): 作者采纳建议, 在 `runner._run` 中添加 `post_step_fn` 参数, 测试改用 `runner.run()`。
 - 移除冗余 `side_effect` 重置 (style): 作者移除重复的 `side_effect` 设置, 并添加注释说明。
 - 适配非阻塞 `drain` (#45595) 后的断言调整 (correctness): 作者 `rebase` 后修改测试, 使用 `block-reuse fence flush` 替代 `all-finished flush`, 并添加 `multi-job fence` 测试和 `req_status` 清理检查。

风险与影响

- 风险: 测试覆盖了内部数据结构 `_block_id_to_pending_jobs`, 若未来重构该字段名称或语义, 测试需要同步更新。但测试通过 `post_step_fn` 保持相对松散的耦合。没有产品代码变更, 引入回归风险极低。
- 影响: 仅影响测试套件, 无功能或用户体验变化。提高 KV 卸载调度器的测试覆盖率, 增强对关键 `fence` 机制的信心。
- 风险标记: 测试依赖内部实现细节, 无自动端到端测试覆盖

关联脉络

- PR #42761 [Bug]: `kv_cache_offloadig` crashes on 0.21.0 - `KeyError` in `self._block_id_to_pending_jobs[bid]`: 本 PR 添加的回归测试直接关联该 issue 中报告的 `KeyError` 崩溃
- PR #42959 [BugFix][kv_offload]: Prevent offloading stale sliding window blocks: 该 PR 修复了 issue #42761, 本 PR 为其添加回归测试

- PR #45595 （未在列表中，但已知为非阻塞 drain 的 PR）：适配非阻塞 drain 后调整测试断言