PR #37051 完整报告

vllm-project/vllm

Fix priority preemption regression test in scheduler

合并时间：2026-04-01 11:36

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/37051

执行摘要

此 PR 修复了 vLLM 调度器中一个之前被跳过的优先级抢占回归测试，通过重写测试函数为确定性多步验证，确保在 KV 块压力下低优先级请求被抢占而高优先级请求保持运行，提高测试可靠性和可维护性，对生产代码无直接影响。

功能与动机

动机源于旧测试依赖重复请求 ID 而失效，且期望立即抢占不符合实际调度行为。PR body 指出："根因是旧测试期望在单个调度步骤后立即抢占，而实际抢占发生在运行请求推进并请求额外 KV 块之后。"因此需要替换测试以更准确验证优先级抢占逻辑。

实现拆解

改动集中于 tests/v1/core/test_scheduler.py 文件的 test_priority_scheduling_preemption 函数：

- 移除跳过注解：删除 @pytest.mark.skip("needs investigation")，使测试重新启用。
- 重写测试逻辑：
  - 添加分阶段注释（Phase 1-3），模拟低优先级请求先运行、高优先级请求后到达的场景。
  - 使用块对齐令牌数（block_size * 2 = 32 tokens）确保每个请求初始占用 2 个块，通过解码步骤触发额外块需求。
  - 移除循环结构，采用扁平化调用 schedule() 和 update_from_output()，精确控制抢占触发点。
- 断言验证：python assert lo1.status == RequestStatus.PREEMPTED # 低优先级请求被抢占 assert hi1 in scheduler.running # 高优先级请求保持运行

评论区精华

review 讨论中核心交锋包括：

> orozery："为什么需要循环 8 次？我们应精确预测逐出。" ezylopx5：调整为动态计算循环边界（tokens_to_next_block + 2），基于块分配数学提升精确性。
>
> orozery："测试应模拟高优先级请求在低优先级之后到达，以验证抢占逻辑。" ezylopx5：重写测试为先运行两个低优先级请求，后加入高优先级请求，确保抢占发生时高低优先级请求同时运行，验证调度器优先抢占最低优先级请求。

风险与影响

风险：

- 回归风险：测试变更可能意外破坏其他测试，但通过精确模拟和断言降低了风险。
- 兼容性风险：无，仅影响测试代码。 影响：
- 对用户无直接影响，这是内部测试改进。
- 提升调度器优先级抢占逻辑的测试覆盖率，增强系统稳定性。
- 团队受益于更健壮的回归测试，减少未来调试成本。

关联脉络

此 PR 与历史 PR #37067 相关，原始 PR 被拆分为调度器测试修复和 CPU 平台检测修复两个部分。从近期历史 PR 看，vLLM 项目持续优化测试和调度器模块（如 #37160 引入 CPU KV 缓存卸载），此 PR 是测试维护的一部分，有助于确保核心调度逻辑在演进中保持正确性。