PR #46595 完整报告

vllm-project/vllm

[Bugfix][MooncakeStore] track resumed requests via scheduler's resumed_req_ids

合并时间：2026-06-25 07:50

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46595

执行摘要

- 一句话：修复 MooncakeStore 抢占恢复后的块表错乱
- 推荐动作：该 PR 修复了一个隐蔽的同步缺陷，代码简洁且配合完善测试，建议精读 build_connector_meta 中 append/replace 分支的逻辑变更。

功能与动机

在混合 KV 缓存（hybrid KV cache）的重度 warm-cache 运行中， MooncakeStore 的 send-thread 抛出 list index out of range / Store chunk out of range 错误。根因是请求被抢占恢复后， _preempted_req_ids 未被清除，导致连接器误用替换语义，而 token_len 仍反映完整长度，造成索引超界。

实现拆解

1. 移除本地 _preempted_req_ids 状态：在 MooncakeStoreScheduler.__init__ 和 build_connector_meta 中删除对 _preempted_req_ids 的初始化、更新和清理逻辑（文件 scheduler.py）。
2. 改用调度器的 resumed_req_ids：在 build_connector_meta 中，将 if req_id in self._preempted_req_ids 替换为 if req_id in cached_reqs.resumed_req_ids，直接使用调度器输出的信息判断块表是替换还是追加（文件 scheduler.py）。
3. 改进异常日志：在 worker.py 的 BaseSendingThread.run 中，将 logger.error 替换为 logger.exception，并提取 req_id 一并输出，便于定位问题请求。
4. 测试配套：新增两个回归测试：test_running_request_not_in_resumed_req_ids_appends_blocks 验证非恢复请求追加块，test_resumed_request_in_resumed_req_ids_replaces_blocks 验证恢复请求替换块。同时调整现有测试用例，移除对 _preempted_req_ids 的依赖，在 scheduler_output 中增加 resumed_req_ids 字段。

关键文件：

- vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 MooncakeStoreScheduler.init, MooncakeStoreScheduler.build_connector_meta）：核心修复：移除本地 _preempted_req_ids，改用调度器的 resumed_req_ids 判断块表替换 / 追加。
- vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/worker.py（模块 工作线程；类别 source；类型 core-logic；符号 BaseSendingThread.run）：改进异常日志，使

用 logger.exception 并输出 req_id，便于调试。

- tests/v1/kv_connector/unit/test_mooncake_store_scheduler.py（模块 测试；类别 test；类型 test-coverage；符号 test_running_request_not_in_resumed_req_ids_appends_blocks, test_resumed_request_in_resumed_req_ids_replaces_blocks）：新增两个回归测试，验证 append 和 replace 分支的正确性；调整现有测试移除对 _preempted_req_ids 的依赖。

关键符号：MooncakeStoreScheduler.build_connector_meta, BaseSendingThread.run

关键源码片段

vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/scheduler.py

核心修复：移除本地 _preempted_req_ids，改用调度器的 resumed_req_ids 判断块表替换 / 追加。

```
class MooncakeStoreScheduler:
    def __init__(self, ...):
        # ... 其他初始化
        self.load_specs: dict[str, LoadSpec] = {}
        self._request_trackers: dict[str, RequestTracker] = {}
        # self._preempted_req_ids: set[str] = set() # 已移除
        self._unfinished_requests: dict[str, ...] = {}
        self._unfinished_request_ids: set[str] = set()

    def build_connector_meta(self, scheduler_output: SchedulerOutput) -> KVConnectorMetadata:
        # ... 清理 finished 等
        preempted_ids = scheduler_output.preempted_req_ids or set()
        # self._preempted_req_ids.update(preempted_ids) # 已移除
        for req_id in preempted_ids:
            self.load_specs.pop(req_id, None)
            if request_tracker := self._request_trackers.get(req_id):
                request_tracker.reset()
            self._unfinished_requests.pop(req_id, None)
        # ... 处理 cached_reqs
        for req_id, new_block_ids in zip(cached_reqs.req_ids, cached_reqs.new_block_ids):
            # ... 跳过 prefill 等
            # 原来：if req_id in self._preempted_req_ids:
            if req_id in cached_reqs.resumed_req_ids:
                # 抢占恢复：替换块表（new_block_ids 是完整表）
                new_block_ids = tuple(b.copy() for b in new_block_ids)
                # self._preempted_req_ids.discard(req_id) # 已移除
            else:
                # 普通 decode：追加块
                pass
```

vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/worker.py

改进异常日志，使用 logger.exception 并输出 req_id，便于调试。

```
class BaseSendingThread(threading.Thread):
```

```
    # ...
    def run(self):
        self.ready_event.set()
        while True:
            request_data = None # 预先初始化
            try:
                request_data = self.request_queue.get()
                if request_data is None:
                    logger.warning("Received a None request!")
                    self.request_queue.task_done()
                    continue
                self._handle_request(request_data)
            except Exception:
                req_id = getattr(request_data, "req_id", "<unknown>")
                logger.exception("Error in %s (req=%s)", self.name, req_id)
```

tests/v1/kv_connector/unit/test_mooncake_store_scheduler.py

新增两个回归测试，验证 append 和 replace 分支的正确性；调整现有测试移除对 _preempted_req_ids 的依赖。

```
  def test_running_request_not_in_resumed_req_ids_appends_blocks():
      """非恢复请求的 new_block_ids 应追加到 tracker，而非替换。"""
      scheduler = _make_bare_scheduler()
      _add_unfinished_request(scheduler, token_ids=list(range(48)),
                              block_hashes=[b"h0", b"h1"], prefill_end_tokens=48)
      # 调度器输出：resumed_req_ids 为空
      meta = scheduler.build_connector_meta(_make_scheduler_output(scheduled_spec_tokens=
      None))
      req_meta = meta.requests[0]
      # 追踪器中原有 [0,1]，追加 [2] →合并为 [0,1,2]
      tracker = scheduler._request_trackers["req-0"]
      assert tracker.allocated_block_ids == ([0, 1, 2],), "应追加新块"
      # token_len（44） <= 块表长度（3*16=48），不会越界
      assert tracker.token_len <= len(tracker.allocated_block_ids[0]) * scheduler._block_size


  def test_resumed_request_in_resumed_req_ids_replaces_blocks():
      """恢复请求的 new_block_ids 应完全替换 tracker 中的旧块。"""
      scheduler = _make_bare_scheduler()
      _add_unfinished_request(scheduler, token_ids=list(range(48)),
                              block_hashes=[b"h0", b"h1"], prefill_end_tokens=48)
      # 调度器输出：resumed_req_ids 包含 "req-0"，new_block_ids 为完整表
      scheduler_output = _make_resumed_scheduler_output(num_scheduled_tokens=0)
      meta = scheduler.build_connector_meta(scheduler_output)
      tracker = scheduler._request_trackers["req-0"]
      assert tracker.allocated_block_ids == ([0, 1, 2],), "应替换为新块"
      # token_len 应与块表匹配
      assert tracker.token_len <= len(tracker.allocated_block_ids[0]) * scheduler._block_size
```

评论区精华

Dao007forever 指出这并非首次出现调度器与 KV 调度器状态不同步的问题，建议审查其他状态，并特别强调 MRv2 中 _preempted_req_ids 未被更新的问题——njhill 回应表示该修复仅适用于 MRv1，MRv2 中将恢复请求视为新请求。

- 状态同步问题 (design): 该 PR 仅修复了 MooncakeStore 连接器的 _preempted_req_ids 不同步问题，剩余状态问题待后续审查。
- MRv2 兼容性 (question): 确认修复范围限于 MRv1。

风险与影响

- 风险：低风险：核心改动是移除一个与调度器状态脱节的本地集合，改为读取调度器直接提供的权威字段 resumed_req_ids，逻辑上更可靠。但需确保所有依赖 _preempted_req_ids 的路径已完全清理（本 PR 已处理）。异常日志改进仅增强可观测性，无副作用。
- 影响：直接影响 MooncakeStore 连接器在抢占恢复场景下的正确性：修复后恢复请求的块表与 token_len 一致，消除索引越界异常。对正常运行无影响。测试覆盖增强，降低回归风险。
- 风险标记：核心路径变更，需要 MRv2 审查

关联脉络

- PR #46363 [KV Offloading] Replace boolINone lookup return with LookupResult enum: 同一模块（MooncakeStore）的近期重构，涉及调度器状态管理。
- PR #45850 [KV Offload] Use background thread for mmap / cpu_tensors pinning: 同一子系统的性能优化，涉及工作线程模式。
- PR #46636 [ROCm] Begin Deprecation Window for CUDA_VISIBLE_DEVICES on ROCm: 无直接关联，仅同仓库近期 PR。