

PR #50984 完整报告

vllm-project/vllm

[Bugfix][Mooncake] Report failed remote KV loads to the scheduler

合并时间: 2026-09-14 17:55

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50984>

执行摘要

- 一句话: Mooncake 解码侧上报失败 KV 加载, 解除请求永久滞留
- 推荐动作: 值得精读, 尤其是负责 P/D 分离、KV 连接器与调度器失败策略的工程师。重点关注三个设计决策: (1) 把「失败」也写入 `finished_recving_reqs`, 用同一出口解除 `WAITING_FOR_REMOTE_KVS` 等待, 这是最小侵入的解法; (2) 用 `PullReqMeta.failed + queue.Queue` 解决多 P 回报顺序与跨线程传递, 而非引入锁; (3) 对空 `local_block_ids` 的显式跳过, 避免触发调度器 `assert req_id in self.requests`, 这是只有在理解 `notify_kv_transfer_request_rejected` 路径后才会想到的边界。阅读时应把它与 NIXL 的 `_handle_failed_transfer()` 对照看, 并注意本 PR 未新增单测: 如果需要在在此基础上改动, 建议先为 `_handle_failed_recv` 的幂等、空 block 与 `process_pulling_result` 的 failed 分支补测试。

功能与动机

Issue#50719 描述生产 P/D 分离场景 (GLM-5.2-NVFP4 + MooncakeConnector + `kv_load_failure_policy='fail'`) 中 D 节点系统性 stall: 日志里 `pulling kv_caches ... failed: Timeout waiting for P side ready` 在一次抓取中出现 516 次, 同时 `deferred` (即 `num_skipped_waiting_reqs`) 持续累积, 请求既不完成也不释放 KV。PR body 指出根因: 请求只能通过 `finished_recving_reqs` 离开 `WAITING_FOR_REMOTE_KVS`, 而失败路径只写了日志; 并且 `kv_load_failure_policy` 是 no-op, 因为 `MooncakeConnector` 从未实现 `get_block_ids_with_load_errors()`。作者明确对齐 `KVConnectorBase_V1` 的约定与 `NixlConnectorWorker._handle_failed_transfer()` 的既有做法。

实现拆解

1. 补齐调度器契约 (入口): 在 `MooncakeConnector` 上新增 `get_block_ids_with_load_errors()`, 断言 `connector_worker` 存在后转发到 `worker`。这是整个修复的开关——此前缺失该方法使 `kv_load_failure_policy` (`fail / recompute`) 完全失效, 调度器拿不到失败 block。
2. 建立失败收集通道: `MooncakeConnectorWorker.__init__` 新增 `self._invalid_block_ids: queue.Queue[set[int]]`。注释写明「Written from the receiver loop, drained from the worker thread」, 选 `queue.Queue` 而非 `set` 是因为写入方是 receiver 协程 / 线程、读取方是 `EngineCore` 的 worker 线程, 随后 `get_block_ids_with_load_errors()` 用 `get_nowait() + queue.Empty` 循环排空。

3. 新增统一失败处理 `_handle_failed_recv(pull metas, req_ids, reason)`: 遍历请求, 跳过 `pull_meta` is None 或已置 failed 的项; 置位 `PullReqMeta.failed` (新增字段, 默认 False); 调用 `self.xfer_stats.record_failed_recv()`; 把 `local_block_ids` (#40860 之后为 `list[list[int]]`) 展平为无效 block 集合后入队; 并把 `pull_meta.d_req_id` 加入 `finished_recving_reqs`——后者是请求离开 `WAITING_FOR_REMOTE_KVS` 的唯一途径, 也是本次修复能解 stall 的关键。
4. 收敛所有失败来源: `receive_kv_from_single_worker()` 中 `MooncakeXferResponseStatus.ERROR` 分支与 `except Exception` 分支由「`logger.error + record_failed_recv + return`」改为调用 `_handle_failed_recv`; `process_pulling_result()` 中 `response.err_reqs` 分支同样改走该方法; `handle_new_engine_id()` 的 bootstrap 失败 (找不到 `remote_engine_id`) 也改为上报, 而不是只打日志。作者的取舍是: 不再额外加 D 侧过期定时器, 因为丢响应已由既有的 `zmq.RCVTIMEO` (`VLLM_MOONCAKE_ABORT_REQUEST_TIMEOUT + 60` 秒) 覆盖。
5. 两处防误伤保护: 其一, 展平后 `invalid` 为空时跳过上报, 注释解释了这类 pull 只是请 P 释放 block (对应 `AsyncLLM.notify_kv_transfer_request_rejected` 提交的 `abort_immediately` 请求), 上报会触发调度器的 `assert req_id in self.requests`; 其二, `process_pulling_result()` 处理 `ok_reqs` 时若 `pull_meta.failed` 为真则 `continue`, 避免「一个 P 失败、另一个 P 成功」把请求重新算作接收完成。
6. 测试与配套: `changed_files_count` 为 1, 没有新增或修改测试文件; PR body 说明本地 `pytest tests/v1/kv_connector/unit/test_mooncake_connector.py` 26 项通过、`ruff-check/ruff-format` 通过, 并在 2x RTX 5090、Qwen3-0.6B、1P1D、`VLLM_MOONCAKE_ABORT_REQUEST_TIMEOUT=30` 上做了对照 E2E (deferred 由 30s 时 2 个降为 0, KV block 释放, 客户端收到 HTTP 500)。无模型精度评估, 也没有配置、schema 或部署脚本改动。

关键文件:

- `vllm/distributed/kv_transfer/kv_connector/v1/mooncake/mooncake_connector.py` (模块 KV 传输; 类别 source; 类型 core-logic; 符号 `_handle_failed_recv`, `get_block_ids_with_load_errors`, `process_pulling_result`, `receive_kv_from_single_worker`): 唯一改动文件。新增 `_handle_failed_recv()` 与 `get_block_ids_with_load_errors()`, 给 `PullReqMeta` 加 failed 去重位, 把 `MooncakeConnector` 的失败 block 上报接口补全, 并把 `ERROR` 响应、`ZMQ` 异常、`err_reqs`、bootstrap 失败四条路径统一接入, 使调度器能执行 `kv_load_failure_policy`。

关键符号: `MooncakeConnectorWorker._handle_failed_recv`,
`MooncakeConnectorWorker.get_block_ids_with_load_errors`,
`MooncakeConnector.get_block_ids_with_load_errors`,
`MooncakeConnectorWorker.process_pulling_result`,
`MooncakeConnectorWorker.receive_kv_from_single_worker`,
`MooncakeConnectorWorker.handle_new_engine_id`

关键源码片段

vllm/distributed/kv_transfer/kv_connector/v1/mooncake/mooncake_connector.py

唯一改动文件。新增 `_handle_failed_recv()` 与 `get_block_ids_with_load_errors()`，给 `PullReqMeta` 加 `failed` 去重位，把 `MooncakeConnector` 的失败 block 上报接口补全，并把 `ERROR` 响应、`ZMQ` 异常、`err_reqs`、`bootstrap` 失败四条路径统一接入，使调度器能执行 `kv_load_failure_policy`。

```
# vllm/distributed/kv_transfer/kv_connector/v1/mooncake/mooncake_connector.py
import queue
from collections.abc import Collection

@dataclass
class PullReqMeta:
    d_req_id: ReqId
    transfer_id: TransferId
    # 按 KV cache group 分组，上报给调度器前需要展平。
    local_block_ids: list[list[int]]
    remote_engine_id: EngineId
    remote_bootstrap_addr: str
    expire_time: float = float("inf")
    # 一个 D 可能同时对接多个 P，用计数跟踪未完成的 pull 任务。
    pull_tasks_count: int = 0
    # 新增：任一 worker 上报失败即置位，避免同一请求后续的「成功」回报覆盖失败状态。
    failed: bool = False

class MooncakeConnectorWorker:
    def __init__(self, ...):
        ...
        self.finished_sending_reqs: set[ReqId] = set()
        self.finished_recving_reqs: set[ReqId] = set()
        # 新增：由 receiver 协程写入、由 worker 线程排空，必须使用线程安全的 queue.Queue。
        self._invalid_block_ids: queue.Queue[set[int]] = queue.Queue()

    def _handle_failed_recv(
        self,
        pull metas: dict[ReqId, PullReqMeta],
        req_ids: Collection[ReqId],
        reason: str,
    ) -> None:
        """Report a failed remote KV load so the scheduler can fail or recompute it."""
        failed: list[ReqId] = []
        for req_id in req_ids:
            pull_meta = pull_metas.get(req_id)
            # 幂等：已标记失败的请求直接跳过，避免重复上报、重复计数。
            if pull_meta is None or pull_meta.failed:
                continue
```

```

pull_meta.failed = True
failed.append(req_id)
self.xfer_stats.record_failed_recv()

# local_block_ids 现在是 list[list[int]] (#40860 之后多 KV cache group) , 统一展平。
invalid = {b for group in pull_meta.local_block_ids for b in group}
if not invalid:
    # 空 block 的 pull 只是请 P 释放 block (请求从未进入调度器, 见
    # AsyncLLM.notify_kv_transfer_request_rejected 提交的 abort_immediately 请求) 。
    # 此时 D 侧没有请求在等 KV 加载, 上报反而会触发调度器的
    # assert req_id in self.requests。
    continue
self._invalid_block_ids.put(invalid)
# 关键: 失败也算「接收结束」, 请求才有可能离开 WAITING_FOR_REMOTE_KVS。
self.finished_recving_reqs.add(pull_meta.d_req_id)

if failed:
    logger.error("pulling kv_caches for %s failed: %s", failed, reason)

def get_block_ids_with_load_errors(self) -> set[int]:
    """Drain the blocks whose remote KV load failed since the last call."""
    result: set[int] = set()
    while True:
        try:
            result.update(self._invalid_block_ids.get_nowait())
        except queue.Empty:
            break
    return result

def process_pulling_result(self, response: MooncakeXferResponse, pull metas) -> None:
    ok_reqs: list[ReqId] = response.ok_reqs or []
    for req_id in ok_reqs:
        pull_meta = pull_metas[req_id]
        if pull_meta.failed:
            # 该请求已被判定失败, 成功回报不再参与 pull_tasks_count 归零。
            continue
        pull_meta.pull_tasks_count -= 1
        if pull_meta.pull_tasks_count == 0:
            self.finished_recving_reqs.add(pull_meta.d_req_id)

    if response.err_reqs:
        # 原先只打日志; 现在与 ZMQ、异常路径统一走失败上报。
        self._handle_failed_recv(
            pull_metas, response.err_reqs, response.err_msg or "unknown error"
        )

class MooncakeConnector(KVConnectorBase_V1, SupportsHMA):
    # 本 PR 补上的关键接口: 调度器靠它拿到失败 block 并套用 kv_load_failure_policy
    # (fail / recompute) 。此前 MooncakeConnector 未实现该方法, 导致该配置项一直是 no-op。

```

```

def get_block_ids_with_load_errors(self) -> set[int]:
    """Get block IDs whose remote KV load failed."""
    assert self.connector_worker is not None
    return self.connector_worker.get_block_ids_with_load_errors()

async def receive_kv_from_single_worker(self, pull metas, req_ids, worker_addr) -> None:
    try:
        with make_zmq_socket(
            self.async_zmq_ctx, worker_addr, zmq.DEALER, bind=False, linger=0
        ) as sock:
            # 丢响应由既有的 RCVTIMEO 兜底，因此本 PR 不再额外加 D 侧过期定时器。
            sock.setsockopt(
                zmq.RCVTIMEO, (envs.VLLM_MOONCAKE_ABORT_REQUEST_TIMEOUT + 60) *
                1000
            )
            await sock.send(encoded_data)
            while True:
                ret_msg = await sock.recv()
                response = self.xfer_resp_decoder.decode(ret_msg)
                if response.status == MooncakeXferResponseStatus.ERROR:
                    # ERROR 响应不携带 err_reqs，因此直接以本批次 req_ids 作为失败集合。
                    self._handle_failed_recv(
                        pull_metas, req_ids, response.err_msg or "transfer error"
                    )
                return
            self.process_pulling_result(response, pull_metas)
            if response.status == MooncakeXferResponseStatus.FINISH:
                break
    except zmq.ContextTerminated:
        logger.debug("ZMQ context terminated, exiting Mooncake receiver thread.")
    except Exception as e:
        # 原来只记录日志并 return，请求会永远停在 WAITING_FOR_REMOTE_KVS。
        self._handle_failed_recv(pull_metas, req_ids, f"transfer failed: {e}")
    return

```

评论区精华

本次 review 的高价值信息几乎全部来自 PR body 而非行级评论（review_comments_count = 0）：

- 作者主动交代与 #36014 的重叠并列本版本的差异（无 D 侧额外过期定时器、额外处理 MooncakeXferResponseStatus.ERROR、bootstrap 失败路径、空 block 保护、local_block_ids 展平、_invalid_block_ids 用 queue.Queue），并明确表态「Happy to fold this into #36014 instead if @machov or @dtcccc prefer」。
- @dtcccc 直接给出 "LGTM! @NickLucche please review and trigger ci when you are ready."，未要求折叠到 #36014。
- 作者连续 ping 了 NickLucche、ivanium、orozery、zhewenl、ZJY0516；最终 @ivanium 以 APPROVED 收尾，@zhewenl 与 @ZJY0516 先后 /ci run，CI 通过后由 ZJY0516 合入。

- claude[bot] 说明「This pull request is from a fork — automated review is disabled」, cudarabbitai[bot] 也报告 no actionable comments, 因此这条核心解码路径的改动实际没有行级评审记录。
- 代码内注释承载了关键设计理由 (空 block 跳过是为了避免触发调度器 `assert req_id in self.requests`; 失败也要写入 `finished_recving_reqs` 才能离开 `WAITING_FOR_REMOTE_KVS`) , 值得作为阅读入口。
- 与 #36014 的重叠及本版本的实现取舍 (design): @dtcccc 回复 LGTM, 未要求折叠进 #36014; PR 独立合入 main。
- 空 block pull 不得上报, 否则触发调度器断言 (correctness): 以代码内注释固化该边界, review 中无人提出异议。
- 跨线程数据结构与多 P 回报去重 (design): 随 PR 一并合入, 未引发争议。
- 评审覆盖度与 CI 流程 (other): 以人工批准 + CI 全绿方式合入, 核心解码路径缺少逐行评审记录。

风险与影响

• 风险:

1. 行为语义变更 (用户可见) : `get_block_ids_with_load_errors()` 补上后 `kv_load_failure_policy` 从 no-op 变为真正生效, 原先「静默卡死」的请求现在会变成失败 (作者 E2E 观察到 HTTP 500) 。对已经接受旧行为的部署, 错误率指标会出现阶跃式上升, 需要运维侧同步预期。
2. 失败粒度是整请求: `_handle_failed_recv` 一旦处理某 `req_id` 就置 `failed=True` 并把该请求全部 `local_block_ids` 展平上报, 即使只是某个 P/tp rank 的瞬时失败。在 `kv_load_failure_policy=recompute` 下会整请求重算, 代价高于理想的分 block 恢复; 反之不漏报, 优先保证不 stall。
3. 线程安全与队列排空: `_invalid_block_ids` 由 receiver 协程写入、worker 线程排空。若某些调用路径不再周期性调用 `get_block_ids_with_load_errors()`, 队列会随失败持续累积 (长尾内存增长) 。提供的材料未包含调度器侧调用点, 建议确认每步都会 drain。
4. 缺少针对性单测: `changed_files_count=1`, 没有测试文件变更。新增的幂等去重、空 block 跳过、队列 drain、failed 与 `pull_tasks_count` 的交互都只靠既有 26 个单测 + 手工 E2E 覆盖; `process_pulling_result` 中 `continue` 会让 `pull_tasks_count` 不再递减, 属于容易回归的细节分支。
5. 统计口径变化: `record_failed_recv()` 从「每次 worker 级失败记一次」变为「每个失败请求在首次标记时记一次」, 同批次多请求失败会放大计数, 依赖 `xfer_stats` 做告警阈值的看板需要重新标定。
6. 多 P 状态竞争: `if pull_meta.failed: continue` 解决了「先失败后成功」的覆盖问题, 但依赖 `failed` 只在接收协程内单线程置位; 材料中未见对并发写入的额外保护说明。- 影响: 影响面集中在 P/D 分离部署中 `kv_connector='MooncakeConnector'` 的解码侧: 用户侧, 故障请求从「连接悬挂、迟迟不返回」变为明确失败 (HTTP 500) 或按策略重算, D 节点的 `deferred/WAITING_FOR_REMOTE_KVS` 不再无界累积, KV block 能及时释放, 因此是高流量下避免整机 stall 的可用性修复。系统侧, 改动是 Mooncake 连接器内部的失败上报接线, 不涉及 API、序列化协议、模型计算与配置项,

MooncakeXferResponse/MooncakeXferResponseStatus 契约不变，与 NixlConnectorWorker._handle_failed_transfer() 的语义对齐降低了多连接器维护成本。团队侧，这个修复把「失败上报接口必须实现」隐性契约显性化，为其他尚未实现 get_block_ids_with_load_errors() 的连接器提供了参考实现；同时它补齐了 #36014 长期未落地的能力，后续若 #36014 合入需要做实现归并而非功能重复。 - 风险标记：核心解码路径变更，行为语义变更：失败策略由 no-op 转为生效，缺少针对性单测覆盖，跨线程队列依赖调度器排空，失败统计口径变化

关联脉络

- PR #36014 （标题未在上下文中提供）Mooncake 远端 KV 加载失败上报的重叠 PR: PR body 明确说明本 PR 与自 2026-03 起开放、最后代码变更在 2026-03-12 的 #36014 重叠，并逐条列出实现差异（无 D 侧过期定时器、处理 ERROR 与 bootstrap 失败、空 block 保护、展平 local_block_ids、queue.Queue 等），是本 PR 最直接的同功能线关联。
- PR #56317 [Bugfix][NixlPush] Guard _remote_agents read in _do_send_reg_notif (X1): 同期 kv-connector v1 的失败 / 竞态加固类修复，涉及 nixl push_worker/base_worker 的线程安全读写，与本 PR 在「连接器失败路径与跨线程状态」上的问题模式一致；本 PR 也对齐了 NixlConnectorWorker._handle_failed_transfer() 的上报语义。
- PR #56786 [Bugfix][EPD] Preserve media processing options in encoder requests: 同为 P/D 分离（EPD 编码器分离）链路上的连接器修复，并配有针对性的单测；与本 PR 对照可见本 PR 未新增测试文件的覆盖缺口。
- PR #56671 [CI][Test] Mock CPU backend block sizes in kv_connector unit conftest: kv_connector 单元测试基础设施改动，本 PR 依赖的 tests/v1/kv_connector/unit/test_mooncake_connector.py 属于同一测试目录；本 PR 声明的 26 passed 正是在该目录下运行。