

PR #31144 完整报告

sgl-project/sglang

[PD] Prevent decode scheduler from blocking on ZMQ sends to a stalled prefill peer

合并时间: 2026-07-25 12:53

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31144>

执行摘要

- 一句话: 修复 decode 在 prefill 崩溃后因 ZMQ 无超时而永久挂起
- 推荐动作: 本 PR 值得所有从事 PD 部署的开发者精读, 展示了分布式系统中经典死锁的分析步骤与最小化修复方案的设计权衡。

功能与动机

当 prefill 实例因 CUDA OOM 等崩溃时, decode 实例的调度器主循环可能在 ZMQ send_multipart 上永久阻塞, 且心跳线程也会因共享锁死锁, 导致服务器停止响应请求, 即使 prefill 重启也无法恢复 (PR #31144 正文)。根源是 ZMQ PUSH socket 未设置 SNDTIMEO。

实现拆解

1. 公共层基础修改: 在 python/sglang/srt/disaggregation/common/conn.py 的 _connect 类方法中, 为 PUSH socket 设置 zmq.SNDTIMEO, 值来自新增环境变量 SGLANG_DISAGGREGATION_ZMQ_SEND_TIMEOUT (默认 1 秒)。此设置影响所有 PUSH socket (send_metadata、_register_kv_args、_send_abort_notification), 从根本上防止无限阻塞。
2. 注册与发送的异常处理: 在 Mooncake、Mori、nixl 三个后端的 _register_kv_args 和 send_metadata 方法中, 将原有的无保护 send_multipart 包裹 try/except zmq.ZMQError, 失败时记录错误、设置 conclude_state = KVPoll.Failed 并 update_status(KVPoll.Failed), 然后返回 False。
3. 失败不污染连接池: 在 common/conn.py 的 _setup_bootstrap_infos 中, 将 _register_kv_args() 调用提前到缓存 connection_pool 之前, 且仅当返回 True 时才缓存, 失败时直接返回。避免后续请求复用指向死 prefill 的过时连接。
4. 环境变量与文档: 在 python/sglang/srt/envron.py 添加 SGLANG_DISAGGREGATION_ZMQ_SEND_TIMEOUT = EnvInt(1), 并在 docs_new/docs/references/environment_variables.mdx 中记录说明。
5. 测试覆盖: 本次改动未新增专属测试, 但通过现有 CI 中 disaggregation 测试套件 (如 test_disaggregation_basic.py) 覆盖。

关键文件:

- python/sglang/srt/disaggregation/common/conn.py (模块 通信层; 类别 source; 类型 core-logic; 符号 _connect, _setup_bootstrap_infos, _register_kv_args): 核心修改: 添

加 SNDTIMEO、调整注册顺序、基类 `_register_kv_args` 返回 bool

- `python/sglang/srt/disaggregation/mooncake/conn.py` (模块 通信层; 类别 source; 类型 core-logic; 符号 `_register_kv_args`, `send_metadata`) : Mooncake 后端实现: `_register_kv_args` 和 `send_metadata` 增加异常捕获, 返回 bool
- `python/sglang/srt/disaggregation/mori/conn.py` (模块 通信层; 类别 source; 类型 core-logic; 符号 `_register_kv_args`, `send_metadata`) : Mori 后端: 类似异常捕获, `_register_kv_args` 返回 bool
- `python/sglang/srt/disaggregation/nixl/conn.py` (模块 通信层; 类别 source; 类型 core-logic; 符号 `_register_kv_args`, `send_metadata`) : nixl 后端: 同样增加异常捕获和返回 bool
- `python/sglang/srt/envron.py` (模块 配置; 类别 source; 类型 configuration) : 新增环境变量 `SGLANG_DISAGGREGATION_ZMQ_SEND_TIMEOUT`

关键符号: `_register_kv_args`, `send_metadata`, `_connect`, `_setup_bootstrap_infos`

评论区精华

关键讨论包括: ShangmingCai 要求最小化改动只保留三个核心点 (SNDTIMEO、环境变量、最小 try/except), SovietPower 采纳并精简代码; `zmq.DONTWAIT` 改为全局 `SNDTIMEO` 并引入独立环境变量; ronhuafeng 通过 fault injection 发现 `_register_kv_args` 失败后返回 True 导致后续请求误用连接池索引, SovietPower 修正为返回 bool 并调整缓存顺序; ShangmingCai 提醒在 except 块中设置 `conclude_state`, 已补充。

- ZMQ 发送超时策略: DONTWAIT vs SNDTIMEO (design): 采用 SNDTIMEO 且默认 1s, 用户可根据网络情况调大。
- PR 范围控制: 最小化改动 (design): SovietPower 精简 PR, 仅保留核心三处修改。
- `_register_kv_args` 返回值与连接池缓存时序 (correctness): 改为返回 bool; 公共层 `_setup_bootstrap_infos` 先调用注册, 成功后才缓存连接池。同时 except 块中设置 `conclude_state` 和 `update_status(Failed)`。
- `_register_kv_args` except 块缺省 `conclude_state` (correctness): SovietPower 确认已添加。

风险与影响

- 风险: 超时阈值风险: 默认 1 秒在极端网络延迟或大消息场景下可能误判, 但正常发送毫秒级, 用户可调大环境变量。行为变更: `_register_kv_args` 失败后不再继续, 修正了之前的隐含 bug。缺少专项测试: 未新增单元测试覆盖超时恢复路径, 回归风险依赖集成测试。未来 ZMQ 配置再次修改时需保持统一原则。
- 影响: 用户影响: 所有使用 PD 部署的用户受益于更高稳定性, decode 实例不再因 prefill 意外崩溃而永久不可用。系统影响: 调度器主循环不再被 ZMQ 发送阻塞, 心跳线程可正常清理, 实例自恢复时间从分钟级降至秒级。团队影响: 统一 ZMQ 超时处理模式, 降低后续排查和维护成本。
- 风险标记: 核心调度路径变更, 缺专项测试覆盖, 依赖超时默认值

关联脉络

- PR #21837 [related] abort → decode unhealthy (mentioned in PR body, not strictly closed): PR 正文指出该问题可能解释了 #21837 中 abort 导致 decode 不健康的现象；共享相同的 socket 锁死锁路径。
- PR #32379 Fix SWA admission livelock on cached-prefix resumes: 同属调度死锁修复方向，展示了 PD 场景中调度健壮性的持续改进。