

PR #29570 完整报告

sgl-project/sglang

Fix disaggregation receiver ZMQ cleanup

合并时间: 2026-06-29 16:22

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29570>

执行摘要

- 一句话: 修复 disaggregation receiver ZMQ socket 未清理
- 推荐动作: 值得精读, 体现了分布式系统中 socket 生命周期管理的设计权衡: 通过 ZeroMQ 选项实现故障快速发现与资源回收。但必须理解 RECONNECT_IVL=-1 配合 SNDTIMEO 的必要性, 否则可能导致整个调度循环卡死。建议与后续修复 #31025 一并阅读。

功能与动机

关联 Issue #28596 描述了 prefill 节点重启时因 ZMQ socket 未清理导致端口与 NCCL 冲突。解码端心跳失败处理虽清理了元数据, 但未关闭或移除缓存的 ZMQ PUSH socket (`CommonKVReceiver._socket_cache`), 造成端口泄露。

实现拆解

1. 修改 `CommonKVReceiver._connect()` 方法: 在新建 ZMQ PUSH socket 后增加两个 `socket.setsockopt` 调用——`zmq.RECONNECT_IVL=-1` (禁用 ZeroMQ 自动重连, 避免死节点被无限重试) 和 `zmq.LINGER=0` (关闭时立即返回, 不等待未发送消息)。
2. 范围精简: 应维护者要求移除了原方案中的 `SNDTIMEO`、`TCP_KEEPALIVE` 等额外选项, 仅保留上述两个核心配置。
3. 保持现有锁机制: `disconnect_endpoint` 方法继续使用每端点锁保护 `sock.close()`, 未采纳移除锁的建议。
4. 移除单元测试: 维护者认为无需独立测试, 测试文件被回滚。

关键文件:

- `python/sglang/srt/disaggregation/common/conn.py` (模块 网络层; 类别 source; 类型 core-logic; 符号 `_connect`): 核心变更文件, 在 `_connect` 方法中新增两行 socket 选项配置, 直接影响 disaggregation 场景下 ZMP socket 的故障清理行为。

关键符号: `CommonKVReceiver._connect`

关键源码片段

`python/sglang/srt/disaggregation/common/conn.py`

核心变更文件，在 `_connect` 方法中新增两行 socket 选项配置，直接影响 disaggregation 场景下 ZMP socket 的故障清理行为。

```
@classmethod
def _connect(cls, endpoint: str, is_ipv6: bool = False):
    with cls._global_lock:
        if endpoint not in cls._socket_cache:
            sock = cls._ctx.socket(zmq.PUSH)
            if is_ipv6:
                sock.setsockopt(zmq.IPV6, 1)
            # 禁用 ZeroMQ 自动重连，避免死节点占用缓存 socket
            sock.setsockopt(zmq.RECONNECT_IVL, -1)
            # 设置 LINGER=0，确保 close() 立即返回，不等待未发送消息
            sock.setsockopt(zmq.LINGER, 0)
            sock.connect(endpoint)
            cls._socket_cache[endpoint] = sock
            cls._socket_locks[endpoint] = threading.Lock()
        return cls._socket_cache[endpoint], cls._socket_locks[endpoint]
```

评论区精华

- 锁与线程安全（design）：gemini-code-assist 建议在 `disconnect_endpoint` 中不加锁直接关闭 socket，但作者基于 PyZMQ 文档指出 `Socket.close()` 非线程安全，坚持保留每端点锁以避免与 `send_multipart` 并发时的竞争。达成共识，维持原锁。
- 变更范围精简（design）：维护者 ShangmingCai 要求只保留 `RECONNECT_IVL` 和 `LINGER`，移除 `SNDBTIMEO` 及 `TCP keepalive` 选项。作者接受并精简。
- 单元测试必要性（testing）：ShangmingCai 认为无需新增测试，作者删除测试文件。
 - `disconnect_endpoint` 中关闭 socket 的锁使用 (design): 维持现有锁保护，不采纳无锁建议。
 - 移除多余的 socket 选项只保留核心两个 (design): 作者更新代码，仅保留两行配置。
 - 添加单元测试是否有必要 (testing): 测试被移除，不引入。

风险与影响

- 风险：主要风险来自 `RECONNECT_IVL=-1`：当对端 PULL socket 死亡后，PUSH socket 进入 mute 状态，后续 `send_multipart` 若无 `SNDBTIMEO` 会无限阻塞（此 PR 未设置 `SNDBTIMEO`）。该问题在后续 PR #31025 中修复。此外，`LINGER=0` 可能导致未发送消息丢弃，但影响有限。锁保护虽避免竞争，但若 `send_multipart` 长时间阻塞，心跳检测线程也可能因等待锁而延迟。
- 影响：影响范围：仅限 disaggregation 部署模式下 decode 侧的 receiver socket 创建逻辑。对正常请求路径无影响（复用缓存 socket），仅在故障恢复时改变 socket 清理行为。需注意用户如果自定义 ZMQ 选项可能被覆盖。
- 风险标记：`RECONNECT_IVL=-1` 缺 `SNDBTIMEO` 保护，线程竞争风险（锁机制）

关联脉络

- 暂无明显关联 PR