

PR #31592 完整报告

sgl-project/sglang

[BugFix][EPD] Harden zmq_to_scheduler receiver failures; sync error info across TP

合并时间: 2026-07-24 16:37

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31592>

执行摘要

- 一句话: 修复 EPD ZMQ 接收解码失败导致 scheduler 崩溃的问题
- 推荐动作: 值得精读的部分: 异常路径包裹的设计模式、跨进程类型兼容处理。建议为 EPD 添加集成测试覆盖解码失败场景, 确保变更长期可维护。

功能与动机

Receiver decode/assemble errors should fail the request instead of crashing the scheduler, and TP ranks need a consistent encoder error payload for abort.

实现拆解

1. 包裹 ZMQ 解码 / 组装路径: 在 `encode_receiver.py` 的 `_try_recv_mm_data` 中, 将 `safe_pickle_loads` 及后续处理置于 try-except 内, 捕获解码 / 组装异常后设置 `self.status = WaitingImageRequestStatus.FAIL`、记录错误消息、清理 GPU 缓冲区并关闭 socket, 避免异常传播到 scheduler 事件循环。
2. 强制 `error_code` 类型: 在 `EmbeddingData.__init__` 中, 将 `error_code` 通过 `int(error_code) if error_code is not None` 转换为原生 int 类型, 规避 `safe_pickle_loads` 白名单对 `http.HTTPStatus` 的阻塞。
3. 跨 TP 同步错误信息: 新增 `_sync_fail_info_across_tp` 方法, 在检测到编码错误后, 通过 `broadcast_pyobj` 在 TP 组间同步 `error_msg` 和 `error_code`, 确保所有 TP rank 收到一致的错误信息用于后续 abort。
4. 增强状态码解析: 在 `request_receiver.py` 的 `_apply_mm_receiver` 中, 将原本简单的 `error_code == 400` 判断改为支持 `None`、`HTTPStatus` 实例或 int 值, 并正确转换为 `HTTPStatus` 用于 `prepare_abort`, 提升健壮性。

关键文件:

- `python/sglang/srt/disaggregation/encode_receiver.py` (模块 EPD 接收器; 类别 source; 类型 core-logic; 符号 `_sync_fail_info_across_tp`, `_try_recv_mm_data`, `EmbeddingData.init`): 核心改动: 包裹 ZMQ 解码 / 组装路径, 新增错误同步方法, 强制 `error_code` 类型。
- `python/sglang/srt/managers/scheduler_components/request_receiver.py` (模块 调度组件; 类别 source; 类型 core-logic; 符号 `_apply_mm_receiver`): 修改状态码解析逻辑, 兼容多种 `error_code` 类型。

关键符号: `_try_recv_mm_data`, `_sync_fail_info_across_tp`, `_apply_mm_receiver`, `EmbeddingData.init`

关键源码片段

`python/sclang/srt/disaggregation/encode_receiver.py`

核心改动: 包裹 ZMQ 解码 / 组装路径, 新增错误同步方法, 强制 `error_code` 类型。

```
# _try_recv_mm_data 中的关键异常处理片段
while self.recv_embedding_data is None or not self.recv_embedding_data.ready:
    try:
        parts = self.recv_socket.recv_multipart(flags=zmq.NOBLOCK, copy=False)
    except zmq.Again:
        # No data available yet, wait a bit and retry
        return
    try:
        recv_obj: EmbeddingData = safe_pickle_loads(parts[0])
        # ... handle error_msg, stale data, reshape embedding ...
    except Exception as e:
        # A message the scheduler cannot decode (blocked unpickle,
        # bad shape/dtype, ...) must fail this request, not crash the
        # scheduler event loop; FAIL still reaches the TP-wide status
        # all-reduce in _process_waiting_requests.
        logger.exception("Failed to decode embedding message for rid=%s", self.rid)
        self.error_msg = f"Failed to decode embedding message: {e}"
        self.status = WaitingImageRequestStatus.FAIL
        self._cleanup_gpu_buffer()
        self.recv_socket.close()
        return

# EmbeddingData.__init__ 中的类型强制转换
# Coerce to plain int: this object crosses process boundaries via
# safe_pickle_loads, whose allowlist blocks http.HTTPStatus.
self.error_code = int(error_code) if error_code is not None else None
```

评论区精华

无实质讨论, 两位 Reviewer (liusy58, ShangmingCai) 均直接 approve, 认可变更。

- 暂无高价值评论线程

风险与影响

- 风险: 主要风险: 1) 新增异常处理路径在正式环境未充分测试, 可能隐藏其他未覆盖错误。 2) `_sync_fail_info_across_tp` 依赖 `broadcast_pyobj` 通信, 若通信本身失败可能导致状态不一致。 3) `error_code` 强制转为 `int` 可能丢失 `HTTPStatus` 的语义信息, 但通过 `request_receiver.py` 中的处理逻辑补偿。整体影响面有限, 仅涉及 EPD 分离模式下的失败路径。

- 影响：影响范围限于 EPD 分离部署模式（encoder_transfer_backend 为 zmq_to_scheduler 或 mooncake）。用户从 scheduler 崩溃变更为请求被优雅 abort，可观测性提升（日志记录错误细节）。系统稳定性增强，正常流程不受影响。
- 风险标记：核心调度路径变更，缺少测试覆盖，跨进程 pickle 兼容性

关联脉络

- PR #31217 [Disagg][StagingBuffer][1/2] Robustness and failure handling: 同属 EPD/disaggregation 稳定性改进系列，增强错误处理路径。