

PR #32071 完整报告

sgl-project/sglang

[BugFix][EPD] Fix Mooncake source-MR lifecycle for multi-TP /send

合并时间: 2026-07-27 15:06

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32071>

执行摘要

- 一句话: 修复 Mooncake 多 TP rank 并发 /send 的内存竞争
- 推荐动作: 建议精读此 PR, 尤其是请求级共享 MR 的 lazy registration 模式和 embedding 生命周期管理, 可作为编码中解决并发资源竞争的范例。

功能与动机

修复 TP>1 时 Mooncake 并发 sibling /send 导致的内存区域重叠错误。PR body 指出: "Each prefill TP rank POSTs /encode + /send for the same req_id. The default batch_encode path never pre-registers a shared MR, so sibling /send calls race on per-call register/deregister and early mm_data.embedding = None. Under concurrency this causes: Transfer Engine does not support overlapped memory region"

实现拆解

1. 请求级共享 MR: 在 _send() 方法中, 首次为某 req_id 调用 /send 时注册 MR, 并将 MR 指针记录在 self._forward_results[req_id]["mr_ptr"] 中; 后续同一 req_id 的 /send 直接复用该 MR, 避免重复 register/deregister。
2. 延迟 MR 注销: 将原先在每次 /send 末尾立即 deregister 的逻辑移除, 改为在 _cleanup_inflight_encode_state() 中统一 deregister, 确保所有 sibling /send 均完成后才释放 MR。
3. 保留 embedding 引用: 删除原来在 /send 末尾 mm_data.embedding = None 的代码; 改为在 MR deregister 之后 (cleanup 函数中) 才将 embedding 和 cached_embedding 置 None, 避免并发 /send 读取到 None。
4. 增强错误上报: 将 transfer_sync 的异步调用改为捕获返回值, 若返回负值则抛出 InternalError, 避免静默失败。

关键文件:

- python/sglang/srt/disaggregation/encode_server.py (模块 调度器; 类别 source; 类型 core-logic; 符号 _send, _cleanup_inflight_encode_state): 核心变更文件, 修改了 Mooncake 后端的 /send 逻辑和 cleanup 函数, 修复并发竞争。

关键符号: _send, _cleanup_inflight_encode_state

关键源码片段

python/sglang/srt/disaggregation/encode_server.py

核心变更文件，修改了 Mooncake 后端的 /send 逻辑和 cleanup 函数，修复并发竞争。

```
# python/sglang/srt/disaggregation/encode_server.py
# 关键变更：请求级共享 MR + 延迟注销
async def _send(
    self,
    req_id: str,
    url: Optional[str] = None,
    session_id: Optional[int] = None,
    buffer_address: Optional[int] = None,
    embedding_port: Optional[int] = None,
    mm_data: EmbeddingData,
    encoder_metrics_collector: Optional[object] = None,
) -> None:
    if get_disagg().encoder_transfer_backend == "mooncake":
        # ... 省略 VIT forward 等待和 embedding 获取 ...

        # 请求级共享 MR，首次 /send 时惰性注册，
        # 注销延迟到 _cleanup_inflight_encode_state 统一处理
        fwd_state = self._forward_results.setdefault(req_id, {})
        mr_already_registered = fwd_state.get("mr_ptr") == embedding.data_ptr()
        if not mr_already_registered:
            self.engine.register(embedding.data_ptr(), embedding.nbytes)
            self._forward_results[req_id]["mr_ptr"] = embedding.data_ptr()

        _t_xfer_start = time.monotonic()
        xfer_ret = await asyncio.to_thread(
            self.engine.transfer_sync,
            session_id,
            embedding.data_ptr(),
            buffer_address,
            embedding.nbytes,
        )
        # 新增：检查 transfer_sync 返回值，失败时抛出异常
        if xfer_ret < 0:
            raise InternalError(
                f"Mooncake transfer_sync failed for {req_id} "
                f"(session={session_id}, nbytes={embedding.nbytes}, "
                f"ret={xfer_ret})"
            )
        # ... 日志和指标 ...

        # ... 后续序列化和发送逻辑 ...

    async def _cleanup_inflight_encode_state(self, req_id: str):
        # ... 取消 background task ...
        mm_data = self.embedding_to_send.pop(req_id, None)
```

```
# 释放 MR (之前是每个 /send 后立即释放)
forward_state = self._forward_results.pop(req_id, None)
if forward_state is not None:
    mr_ptr = forward_state.get("mr_ptr")
    if mr_ptr is not None:
        try:
            self.engine.deregister(mr_ptr)
        except Exception as dereg_err:
            logger.warning(
                f"Shared-MR deregister failed for {req_id}: {dereg_err}"
            )
    forward_state.pop("embedding", None)

# 延迟清除 embedding, 确保所有 /send 完成前引用有效
if mm_data is not None:
    mm_data.embedding = None
    mm_data.cached_embedding = None
self._forward_ready_events.pop(req_id, None)
```

评论区精华

Review 讨论较少，主要集中在 CI 重跑。两个 reviewer 均直接批准，无设计争议。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。改动集中在 Mooncake 后端路径，只影响 encode_server.py 中的 _send 和 _cleanup_inflight_encode_state 函数。主要风险是：若 cleanup 函数因异常未能执行，MR 可能泄漏；但原有代码已有 backstop (send_timeout) 兜底。另外，xfer_ret < 0 的异常抛出可能改变之前静默失败的行为，需确保上游有对应错误处理。
- 影响：影响范围窄，仅修复 Mooncake 在 TP>1 下的并发 bug。对非 Mooncake 后端无影响。性能上无负面变化，但增加了请求级共享 MR 减少了重复注册开销，可能略有提升。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #31591 [EPD] early release mm_data.embedding: PR body 提及 #31591 提供的早期释放机制与此 PR 的延迟释放形成对比，且 PR body 中标注了 'early release via #31591'。