

PR #30222 完整报告

sgl-project/sglang

Fix disagg speculative decoding with NIXL connector

合并时间: 2026-07-07 06:51

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/30222>

执行摘要

- 一句话: 修复 NIXL 连接器解耦推测解码 KV 长度不匹配
- 推荐动作: 建议精读: 展示了如何在破坏现有架构的前提下, 通过长度切片优雅解决数据不对称问题, 并显式禁止未测试路径。值得关注的决策在于对 HiSparse + decode-only spec-dec 组合的保守处理。

功能与动机

PR #27563 后, 解耦推测解码因 decode 侧相比 prefill 侧多了 draft model 的 KV 缓存 ($n_{dst} > n_{src}$), 导致 `_prep_equal_tp_dlist` 中 dlist 几何长度不匹配, 报错 `ValueError: NIXL prepared dlist geometry length mismatch`。PR Body 指出需恢复忽略多余 KV 条目的行为。

实现拆解

1. 在 `_prepare_payload_xfer` 方法中增加几何校验 (conn.py 第 914-921 行): 计算源和目标 KV 条目数, 若 $n_{dst} < n_{src}$ 则视为不可达错误直接抛异常, 否则根据 $n_{dst} > n_{src}$ 标记 `decode_only_spec_dec`。
2. 拒绝 HiSparse + decode-only spec-dec 组合 (第 930-934、943-947 行): 当一个 `NotImplementedError` 分支或同质 VRAM 分支遇到该标记时, 直接抛 `NotImplementedError`, 避免未测试的路径导致静默错误。
3. 切片目标方 KV 数据到匹配长度 (第 965-966 行): 将 `dst_kv_ptrs` 和 `dst_kv_item_lens` 均切片到 n_{src} 长度, 然后按常规路径构建 dlist, 从而忽略 decode-only spec-dec 多出的条目。

关键文件:

- `python/sglang/srt/disaggregation/nixl/conn.py` (模块 连接器; 类别 source; 类型 core-logic; 符号 `_prepare_payload_xfer`): 核心变更文件, 修改了 `_prepare_payload_xfer` 方法, 增加几何校验、切片目标 KV 数据、禁止未测试组合。

关键符号: `_prepare_payload_xfer`

关键源码片段

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

核心变更文件，修改了 `_prepare_payload_xfer` 方法，增加几何校验、切片目标 KV 数据、禁止未测试组合。

```
# python/sglang/srt/disaggregation/nixl/conn.py
# 片段: _prepare_payload_xfer 的核心修复逻辑 (head 版本)

def _prepare_payload_xfer(self, peer_info: KVArgsRegisterInfo):
    assert self.src_mem_kind is not None
    src_mem_kind = self.src_mem_kind

    # 统计源 (prefill) 和目标 (decode) 的 KV 条目数
    n_src = len(self.kv_args.kv_item_lens)
    n_dst = len(peer_info.dst_kv_item_lens)

    # p 端 (decode) 不应比 d 端 (prefill) 条目少; 若发生则是内部逻辑错误
    if n_dst < n_src:
        raise ValueError(
            "NIXL PD transfer: decode registered fewer KV regions "
            f"({n_dst}) than prefill ({n_src}); unexpected geometry"
        )

    # 判断是否属于 decode-only speculative decoding:
    # decode 侧因 draft model 多出一倍 KV 条目
    decode_only_spec_dec = n_dst > n_src

    if self.is_mla_backend or peer_info.decode_tp_size == self.attn_tp_size:
        dst_mem_kind = None
        try:
            dst_mem_kind = _homogeneous_kv_mem_kind(
                peer_info.dst_kv_mem_kinds, "destination"
            )
        except NotImplementedError:
            # HiSparse + decode-only spec-dec 组合尚未实现
            if decode_only_spec_dec:
                raise NotImplementedError(
                    "NIXL PD transfer does not support HiSparse combined with "
                    "decode-only speculative decoding."
                )
            # 原有混合类型处理
            ...
        return

    if decode_only_spec_dec and dst_mem_kind != "VRAM":
        raise NotImplementedError(
            "NIXL PD transfer does not support HiSparse combined with "
            "decode-only speculative decoding."
        )
    ...
```

```

dst_num_slots = ...

# 关键修复：只取与源端相同数量的目标 KV 数据，忽略 decode-only spec-dec 多余部分
dst_kv_ptrs = peer_info.dst_kv_ptrs[:n_src]
dst_kv_item_lens = peer_info.dst_kv_item_lens[:n_src]
dst_kv_data_lens = [
    item_len * dst_num_slots for item_len in dst_kv_item_lens
]

self._init_equal_tp_prep_handle(
    peer_info.agent_name,
    dst_kv_ptrs,
    peer_info.gpu_id,
    num_slots=peer_info.dst_num_slots,
    mem_kind=dst_mem_kind,
    kv_item_lens=dst_kv_item_lens,
    kv_data_lens=dst_kv_data_lens,
    # xfer_lens 保持源端的长度，确保传输条目一致
    kv_xfer_lens=self.kv_args.kv_item_lens,
)
else:
    # 异构 TP 路径不变
    ...

```

评论区精华

reviewer ishandhanani 认为这是个好发现，并提议为 disagg + hisparse + specdec 添加额外测试。hzh0425 同意但要求测试稳定，此前已将一个 nixl+hisparse 测试改为手动触发。ishandhanani 表示本地测试未出现挂死，但为了稳妥，该组合测试保持手动，未来可考虑将 MTP 加入测试。最终 CI 通过后合并。

- 增加 disagg + hisparse + specdec 测试 (test): 不在此 PR 增加测试，保持现有手动测试机制。

风险与影响

- 风险：变更集中在 python/sglang/srt/disaggregation/nixl/conn.py 的 `_prepare_payload_xfer` 方法，涉及解耦场景下 KV 传输的核心逻辑。主要风险在于：
 - 切片操作假设多余 KV 条目总是位于列表末尾，若未来注册顺序变化可能导致切片错误。
 - `decode_only_spec_dec` 的检测依赖长度比较，若 `prefill` 也使用了 `spec-dec`（理论不可达但未来可能变化），则逻辑会误判。
 - 缺少测试覆盖，尤其是 disagg + hisparse + specdec 场景被明确禁止但无回归测试。
 - 影响：影响范围较窄，仅修复 NIXL 连接器在解耦推测解码下的 bug。用户若使用该组合（`--enable-nixl + disagg + speculate decoding`），将从此前报错变为正常工作。对非 NIXL 或不使用 `spec-dec` 的部署无影响。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #27563 (推测) 引入 NIXL disagg KV 传输重构：此 PR 是修复 #27563 引入的 regression。