PR #31901 完整报告

sgl-project/sglang

[HiSparse]Fix DeepSeek V4 HiSparse PD Transfers with Separate Host and Device KV Indices

合并时间：2026-08-03 20:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31901

执行摘要

- 一句话：修复 HiSparse PD 传输双页索引错位
- 推荐动作：值得精读。核心看点在 conn.py 的双套 transfer 计划设计：通过 dst_device_data_ptrs 集合做目标指针到索引空间的映射，避免在热路径上为每个缓冲区分叉；同时 send_kvcache 中基于压缩比推导 C4 层数的方式很轻量但依赖配置准确性。部署 PD + DeepSeek-V4 + HiSparse 的团队应尽快合入此修复，并关注后续 DCP 兼容 PR。

功能与动机

issue #31482 报告 DeepSeek-V4-Flash-FP8 在 PD 分离 + decode 侧 HiSparse 部署下与普通 PD 部署产生不同 logits：GPQA 长推理收敛下降，SWE agent 因 DSML 工具调用标记损坏而循环到 step 上限。PR body 指出：HiSparse 独立分配 host 页和逻辑 device 页，页 ID 不保证一致，原直接到 host 的传输路径把 host KV 页索引同时用于 host 与 device 目标缓冲区，导致 C4 indexer 与 C128 device 缓冲区写错页。

实现拆解

1. decode 侧计算 device 页索引：在 python/sglang/srt/disaggregation/decode.py 中，当 enable_hisparse 开启且池为 DeepSeekV4TokenToKVPool（非 fake transfer）时，从 req_to_token_pool.req_to_token 提取完整的逻辑 KV 索引，经 kv_to_page_indices 转换为 device_page_indices，并通过 send_metadata 的 device_kv_indices 参数携带；非 Mooncake backend 直接抛 NotImplementedError，避免其他后端静默写错页。
2. 扩展 ZMQ 传输协议：python/sglang/srt/disaggregation/mooncake/conn.py 中 TransferInfo 新增 dst_device_kv_indices 字段，from_zmq 解析消息第 10 个字节字段（len(msg) > 9 判断保持向后兼容）；send_metadata 增加 device_kv_indices 参数并在消息尾部追加序列化字节。
3. 双套传输计划：_send_kvcache_generic 新增 dst_device_data_indices 与 dst_device_data_ptrs 参数，同时用 group_concurrent_contiguous 构建 host 与 device 两套 transfer 块，set_transfer_blocks 根据目标指针是否属于 device 缓冲区选择对应块；send_kvcache 在收到 device 索引时，依据 mla_compression_ratios 计算 C4 层数，推导 device 目标指针集合 dst_kv_ptrs[c4_layer_num:]。
4. 分块传输与测试配套：transfer_worker 按 chunk 对 device 索引做切片并传给 send_kvcache，DCP relayout 场景若携带 device 索引直接抛 RuntimeError；测试文件 test/registered/unit/mem_cache/test_hisparse_allocator.py 新增 3 个单测，分别覆盖双

索引传输块选择、PP 分区下 device 指针推导、ZMQ 反序列化，并调整原有 host 分配单测的 page_size 与页归属断言。

关键文件：

- python/sglang/srt/disaggregation/mooncake/conn.py（模块 PD 传输；类别 source；类型 core-logic；符号 _send_kvcache_generic, send_kvcache, TransferInfo, transfer_worker）：核心传输逻辑变更：TransferInfo 协议扩展、双套 transfer 计划构建与选择、send_kvcache 的 device 指针集合推导、transfer_worker 分块适配。
- python/sglang/srt/disaggregation/decode.py（模块 解码侧；类别 source；类型 core-logic；符号 DecodePreallocQueue）：decode 侧在 HiSparse 场景下计算并携带 device 逻辑页索引，且对非 Mooncake 后端加保护。
- test/registered/unit/mem_cache/test_hisparse_allocator.py（模块 内存分配；类别 test；类型 test-coverage；符号 test_mooncake_uses_separate_host_and_device_page_indices, test_mooncake_derives_device_buffers_from_local_pp_layout, test_mooncake_transfer_metadata_carries_device_page_indices）：新增 3 个单测覆盖双索引传输块选择、PP 布局推导和 ZMQ 反序列化，并调整原有 host 分配单测。

关键符号：MooncakeKVManager._send_kvcache_generic, MooncakeKVManager.send_kvcache, TransferInfo.from_zmq, MooncakeKVManager.send_metadata, MooncakeKVManager.transfer_worker, test_mooncake_uses_separate_host_and_device_page_indices, test_mooncake_derives_device_buffers_from_local_pp_layout, test_mooncake_transfer_metadata_carries_device_page_indices

关键源码片段

python/sglang/srt/disaggregation/mooncake/conn.py

核心传输逻辑变更：TransferInfo 协议扩展、双套 transfer 计划构建与选择、send_kvcache 的 device 指针集合推导、transfer_worker 分块适配。

```
def _send_kvcache_generic(
    self,
    mooncake_session_id: str,
    src_data_ptrs: list[int],
    dst_data_ptrs: list[int],
    item_lens: list[int],
    prefill_data_indices: npt.NDArray[np.int32],
    dst_data_indices: npt.NDArray[np.int32],
    executor: concurrent.futures.ThreadPoolExecutor,
    state_type: Optional[StateType] = None,
    force_flat: bool = False,
    src_layer_ids: Optional[List[int]] = None,
    dst_layer_ids: Optional[List[int]] = None,
    dst_device_data_indices: Optional[npt.NDArray[np.int32]] = None,
    dst_device_data_ptrs: Optional[set[int]] = None,
) -> int:
```

```
    # HiSparse 下 host（C4 稀疏）与 device（逻辑页）使用两套页空间，页 ID 不对齐，
    # 因此这里同时构建两套 transfer 计划，后续按目标指针归属选择。
    prefill_kv_blocks, dst_kv_blocks = group_concurrent_contiguous(
        prefill_data_indices, dst_data_indices
    )
    device_prefill_kv_blocks = device_dst_kv_blocks = None
    if dst_device_data_indices is not None:
        device_prefill_kv_blocks, device_dst_kv_blocks = (
            group_concurrent_contiguous(
                prefill_data_indices, dst_device_data_indices
            )
        )

    # ... 中间省略 layers_params 构建（MLA / MHA / PP 相关）...

    def set_transfer_blocks(src_ptr: int, dst_ptr: int, item_len: int):
        transfer_blocks = []
        if dst_device_data_ptrs and int(dst_ptr) in dst_device_data_ptrs:
            # 目标是 C4 indexer / C128 这类 device 缓冲区时，使用逻辑 device 页索引。
            assert (
                device_prefill_kv_blocks is not None
                and device_dst_kv_blocks is not None
            )
            src_blocks, dst_blocks = (
                device_prefill_kv_blocks,
                device_dst_kv_blocks,
            )
        else:
            # 其余 host 缓冲区继续使用 C4 host 页索引。
            src_blocks, dst_blocks = prefill_kv_blocks, dst_kv_blocks
        for prefill_index, decode_index in zip(src_blocks, dst_blocks):
            src_addr = src_ptr + int(prefill_index[0]) * item_len
            dst_addr = dst_ptr + int(decode_index[0]) * item_len
            length = item_len * len(prefill_index)
            transfer_blocks.append((src_addr, dst_addr, length))
        return transfer_blocks

    # ... 后续 process_layer / process_layers 复用 set_transfer_blocks ...
```

python/sglang/srt/disaggregation/decode.py

decode 侧在 HiSparse 场景下计算并携带 device 逻辑页索引，且对非 Mooncake 后端加保护。

```
# 仅当 HiSparse 开启且池是 DeepSeekV4TokenToKVPool 时才需要携带 device 索引。
device_page_indices = None
if (
    self.scheduler.enable_hisparse
    and isinstance(self.token_to_kv_pool, DeepSeekV4TokenToKVPool)
    and not _is_fake_transfer(decode_req.req, self.scheduler.server_args)
):
```

```
        # alloc_logical_only() 已分配 C4 indexer 与 C128 KV 共享的逻辑页，
        # 这些 device 缓冲区不走 C4 稀疏物理槽位映射，
        # 因此把逻辑页 ID 与独立分配的 C4 host 页 ID 一起发送给对端。
        full_kv_indices = self.req_to_token_pool.req_to_token[
            decode_req.req.req_pool_idx,
            prefix_len:origin_input_len,
        ]
        device_page_indices = kv_to_page_indices(
            full_kv_indices,
            page_size,
        ).astype(np.int32)
        if self.transfer_backend != TransferBackend.MOONCAKE:
            # 目前只有 Mooncake 支持双索引直传，其他后端主动报错避免静默写错页。
            raise NotImplementedError(
                "DSV4 HiSparse direct PD transfer currently requires "
                "the Mooncake backend"
            )

    metadata_kwargs = {"decode_prefix_len": total_prefix_len}
    if device_page_indices is not None:
        metadata_kwargs["device_kv_indices"] = device_page_indices

    decode_req.kv_receiver.send_metadata(
        page_indices,
        decode_req.metadata_buffer_index,
        state_indices,
        **metadata_kwargs,
    )
```

评论区精华

审阅者 ShangmingCai 在 review 中提出两点：一是“Probably need to check the diff and compatibilitywiththeDCPthatwasrecentlyadded.”（需要核对与最近新增DCP的兼容性），作者回应“Okay, I'll add support for HiSparse and DCP compatibility in the next PR.”——当前 PR 在 transfer_worker 中对 DCP relayout + device 索引直接抛 RuntimeError，把完整兼容留到后续 PR；二是 approve 时要求 rebase 解锁 CI，作者多次 merge main 并触发 /rerun-failed-ci 使 CI 通过。gemini-code-assist 机器人仅生成摘要，无实质性评论。

- HiSparse 与 DCP relayout 的兼容性 (design): 作者回应将在下一个 PR 中支持 HiSparse 与 DCP 兼容；当前实现中 DCP relayout 遇到 device 索引会抛 RuntimeError，避免静默错误数据。
- rebase 解锁 CI (other): 作者多次 merge main 并触发 /rerun-failed-ci，最终 PR Test 通过，PR Test (Extra) 失败未阻塞合并。

风险与影响

- 风险：

1. DCP 兼容性缺口：transfer_worker 中 DCP relayout 遇到 dst_device_kv_indices 会抛 RuntimeError，未来若合并 DCP 支持需要显式跟进，否则 PD DCP + HiSparse 组合不可用。
2. 仅支持 Mooncake backend：decode.py 对非 Mooncake 后端直接 NotImplementedError，其他传输后端（如 Triton/ 自定义）下 HiSparse PD 直传功能不可用，需确认部署面。
3. device 指针集合推导依赖配置：send_kvcache 中 dst_device_kv_ptrs = set(dst_kv_ptrs[c4_layer_num:]) 依赖 mla_compression_ratios 与 prefill_start_layer/end_layer 的准确性，PP 分区变化或压缩比配置异常时可能误判缓冲区归属。
4. 测试覆盖集中在单测：新增测试均为 mock 级单元测试，GPQA/SWE 精度验证由作者在评论中给出，尚未固化为 CI 回归测试。 - 影响：对用户：修复 DeepSeek-V4 在 PD 分离 + HiSparse 部署下的输出损坏，作者验证 GPQA 精度从 81.63% 恢复到正常区间（三次运行 No HiSparse 86.87% vs HiSparse 87.71%），SWE 工具调用 160/160 有效、0 个畸形调用。对系统：Mooncake 传输元数据新增可选字段，旧消息通过 len(msg) > 9 保持向后兼容；非 HiSparse 路径不受影响。对团队：确立 HiSparse 需要 host/device 双索引直传的设计方向，后续需补充 DCP 兼容。 - 风险标记：核心传输路径变更，DCP 兼容性未覆盖，仅支持 Mooncake backend，缺少端到端回归测试

关联脉络

- PR #33125 [rust-server] PD disaggregation support: 同属 PD 分离部署能力建设，Rust server 端为 PD 提供 KV bootstrap 注册，本 PR 在 Python 侧补齐 HiSparse 双页索引直传。