PR #26539 完整报告

sgl-project/sglang

[PD][MoRI] Align hybrid state transfer with per-component schema

合并时间：2026-05-29 15:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26539

执行摘要

- 一句话：修复 MoRI 后端混合状态传输组件感知架构，消除 PD 引导崩溃
- 推荐动作：值得精读，特别是 conn.py 中组件分发逻辑和序列化方案的设计抉择（pack_int_lists vs 自定义 msgpack）。开发者可以学习如何将一种传输后端与新的架构对齐，以及如何设计兼容旧格式的升级路径。

功能与动机

由于 PR #24932 重构混合状态传输为每组件架构，但 MoRI 的 _register_kv_args、send_state、TransferInfo 等仍使用扁平状态假设，导致 struct.error: required argument is not an integer 崩溃（issue #26525）。此 PR 将 MoRI 与 Mooncake/NIXL 已经使用的每组件调度模型对齐。

实现拆解

1. 新增序列化 / 反序列化辅助函数：在 conn.py 中添加 _normalize_state_indices_per_component、_pack_state_indices、_unpack_state_indices、_pack_mem_desc_lists、_unpack_mem_desc_lists，分别处理组件感知的状态索引和 MemoryDesc 列表的打包 / 解包。新增函数使用 pack_int_lists/unpack_int_lists（从 common.utils 导入）和 msgspec.msgpack 嵌套编码。
2. 升级数据结构定义：TransferInfo.dst_state_indices 从 npt.NDArray[np.int32] 改为 List[npt.NDArray[np.int32]]；KVArgsRegisterInfo.dst_state_mem_descs 从 List[MemoryDesc] 改为 List[List[MemoryDesc]]，dst_state_item_lens 和 dst_state_dim_per_tensor 从 List[int] 改为 List[List[int]]；MoriKVManager.state_mem_descs 也改为 List[List[MemoryDesc]]。
3. 改写 send_state 分发逻辑：从直接使用扁平 state_indices 改为迭代 state_types[i]，对每个组件判断类型（Mamba 或 SWA/DSA）后分派到 _send_mamba_state 或 _send_swa_dsa_state 独立传输。
4. 更新注册和元数据路径：_register_kv_args、send_metadata 等函数适配新格式，使用新打包函数序列化 state_mem_descs 和 state_indices。
5. 测试配套：添加 TestMoriTransferEngineHybridMambaE2E 测试类，继承 MoriTransferEngineBase，使用 DEFAULT_HYBRID_MAMBA_MODEL_NAME_FOR_TEST 模型在 8 GPU 上运行烟雾测试，验证混合状态传输正确性。同时增加模型选择的可扩展

性支持。

关键文件：

- python/sglang/srt/disaggregation/mori/conn.py（模块 传输层；类别 source；类型 core-logic；符号 _normalize_state_indices, _normalize_state_indices_per_component, _pack_state_indices, _unpack_state_indices）：核心文件，实现所有组件感知的序列化、数据结构和传输分发逻辑，是此 PR 的主要变更点。
- test/registered/amd/disaggregation/test_mori_transfer_engine_e2e.py（模块 MoRI 测试；类别 test；类型 test-coverage；符号 TestMoriTransferEngineHybridMambaE2E, test_generate_smoke_hybrid_mamba）：添加混合 Mamba 模型状态传输回归测试，验证组件感知路径的正确性。

关键符号：_normalize_state_indices_per_component, _pack_state_indices, _unpack_state_indices, _pack_mem_desc_lists, _unpack_mem_desc_lists, send_state, _register_kv_args, _send_mamba_state, _send_swa_dsa_state

关键源码片段

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

核心文件，实现所有组件感知的序列化、数据结构和传输分发逻辑，是此 PR 的主要变更点。

```
def _normalize_state_indices_per_component(
    state_indices: Optional[List],
) -> Optional[List[Optional[npt.NDArray[np.int32]]]]:
    # 将每组件状态索引规范化为 ravel 后的数组列表
    if state_indices is None:
        return None
    out: List[Optional[npt.NDArray[np.int32]]] = []
    for entry in state_indices:
        if entry is None:
            out.append(None)
        else:
            out.append(np.asarray(entry, dtype=np.int32).ravel())
    return out


def _pack_state_indices(
    state_indices: Optional[List[Optional[npt.NDArray[np.int32]]]],
) -> bytes:
    # 将组件状态索引列表打包为字节流，使用 pack_int_lists ( 格式 "i")
    if not state_indices:
        return b""
    lists = [(arr.tolist() if arr is not None else []) for arr in state_indices]
    return pack_int_lists(lists, "i")


def _unpack_state_indices(buf: bytes) -> List[npt.NDArray[np.int32]]:
    # 从字节流解包为组件状态索引数组列表
```

```
    if not buf:
        return []
    return [np.asarray(lst, dtype=np.int32) for lst in unpack_int_lists(buf, "i")]


def _pack_mem_desc_lists(mems_per_comp: List[List[MemoryDesc]]) -> bytes:
    # 将每组件 MemoryDesc 列表打包为嵌套 msgpack 字节流
    if not mems_per_comp:
        return b""
    return msgspec.msgpack.encode(
        [[mem.pack() for mem in comp] for comp in mems_per_comp]
    )


def _unpack_mem_desc_lists(blob: bytes) -> List[List[MemoryDesc]]:
    # 从嵌套 msgpack 字节流解包为每组件 MemoryDesc 列表
    if not blob:
        return []
    nested = msgspec.msgpack.decode(blob)
    return [[MemoryDesc.unpack(b) for b in comp] for comp in nested]
```

评论区精华

复审者 ShangmingCai 在 conn.py 的 send_state 函数中建议简化 state_types 获取："state_types = self.kv_args.state_types is enough"，并在另一位置提出类似简化。作者 maning00 接受并更新代码。该讨论确保了代码简洁性，且维持了正确性假设（state_types 必定已设置）。

- 简化 state_types 获取的反馈 (correctness): 作者 maning00 接受建议并更新代码。

风险与影响

- 风险：
  1. 序列化兼容性：state_indices 序列化从 np.frombuffer 改为 pack_int_lists/unpack_int_lists，与其他后端（Mooncake/NIXL）独立，但若未来共享序列化工具需注意对齐。
  2. 性能影响：新增的列表转换（numpy 与 Python 列表互转）在大型状态传输中可能有微小开销，但每次传输仅一次序列化，影响可忽略。
  3. 测试覆盖有限：仅一个混合模型烟雾测试，未覆盖多种头配置或极端大小，但已覆盖基本混合场景。
  4. 状态类型顺序假设：若 state_types 与 indices 顺序不匹配会导致错误传输，但代码确保从同一 kv_args 取值。 - 影响：对用户：修复了使用 MoRI 后端的 PD 引导崩溃，使混合状态模型（如 Qwen3.5、DeepSeek V4）可正常使用。对系统：wire 格式和 API 调整，但与其他传输后端解耦，不会影响 Mooncake/NIXL。对团队：简化 MoRI 代码结构，使其更易于维护，并与其他后端模式统一。 - 风险标记：核心路径变更，序列化兼容性，测试覆盖有限

关联脉络

- PR #24932 [PD] Refactor hybrid state transfer: 该 PR 引入每组件状态架构，此 PR 完成 MoRI 后端的迁移对齐。