PR #23606 完整报告

sgl-project/sglang

[HiSparse & PD] Support hisparse memory pool host page > 1

合并时间：2026-05-19 16:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23606

执行摘要

- 一句话：支持 HiSparse 主机池页大小大于 1
- 推荐动作：建议合并并关注性能基准测试结果，特别是 PD 模式下的吞吐量。设计上的 Mixin 抽象值得后续参考。

功能与动机

Currently, the host pool in hisparse has a hardcoded configuration of page=1. Consequently, page scheduling is interrupted at this point; when hisparse is integrated with hicache—particularly regarding data transfer performance between storage and the host—performance degrades significantly.

实现拆解

1. 在 memory_pool_host.py 中引入 HiSparseHostPoolMixin，提供 alloc_paged_token_slots 和 allocated_host_indices 等页级别分配方法。
2. 修改 MLATokenToKVPoolHost 继承该 Mixin，并将 page_size 参数从硬编码 1 改为从设备池获取。
3. 修改 hisparse_coordinator.py 中的 __init__ 和 admit_request_into_staging、_eager_backup_previous_token，使用新的分配接口。
4. 修改 decode.py 中的 _init_kv_manager 和 _pre_alloc，移除特判分支，统一使用 alloc_paged_token_slots。
5. 移除 mooncake/conn.py 中的 send_kvcache_hisparse 方法及 enable_hisparse 字段，因为页大小统一后不再需要特殊处理。
6. 移除 common/conn.py 中针对 hiSparse 的页大小不匹配绕过逻辑。
7. 新增单元测试 test_single_node_staging_allocates_paged_host_slots 和 test_pd_decode_prealloc_hisparse_host_slots，验证页级分配正确性。

关键文件：

- python/sglang/srt/mem_cache/memory_pool_host.py（模块 主机池；类别 source；类型 dependency-wiring；符号 HiSparseHostPoolMixin, _round_up_to_page_size, alloc_page, alloc_paged_token_slots）：核心变更文件，新增 HiSparseHostPoolMixin 实现页级主机分配，并修改 MLATokenToKVPoolHost 继承该混入类。

- python/sglang/srt/disaggregation/mooncake/conn.py（模块 传输层；类别 source；类型 core-logic；符号 send_kvcache_hisparse）：删除 send_kvcache_hisparse 方法，因为页大小统一后不再需要特殊处理。
- python/sglang/srt/managers/hisparse_coordinator.py（模块 协调器；类别 source；类型 core-logic）：修改协调器，使用新的页级分配接口，增加 allocated_len 跟踪。
- python/sglang/srt/disaggregation/decode.py（模块 解码器；类别 source；类型 core-logic）：解码器侧移除 vs 1 页大小的特判，统一使用 token_to_kv_pool 的 page_size。
- test/registered/unit/managers/test_hisparse_unit.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_single_node_staging_allocates_paged_host_slots, test_pd_decode_prealloc_hisparse_host_slots）：新增页级分配测试用例。
- python/sglang/srt/disaggregation/common/conn.py（模块 通用连接；类别 source；类型 core-logic）：移除 hisparse 页大小不匹配绕过逻辑。
- python/sglang/srt/mem_cache/hisparse_memory_pool.py（模块 稀疏内存池；类别 source；类型 core-logic；符号 DeepSeekV4SingleKVPoolHost）：修改 DeepSeekV4SingleKVPoolHost 的 page_size 初始化，从硬编码 1 改为动态获取。

关键符号：HiSparseHostPoolMixin._round_up_to_page_size, HiSparseHostPoolMixin.alloc_page, HiSparseHostPoolMixin.alloc_paged_token_slots, HiSparseHostPoolMixin.allocated_host_indices, MLATokenToKVPoolHost.init, DeepSeekV4SingleKVPoolHost.init, HiSparseCoordinator.admit_request_into_staging, HiSparseCoordinator._eager_backup_previous_token, Decode._init_kv_manager, Decode._pre_alloc

关键源码片段

python/sglang/srt/mem_cache/memory_pool_host.py

核心变更文件，新增 HiSparseHostPoolMixin 实现页级主机分配，并修改 MLATokenToKVPoolHost 继承该混入类。

```
class HiSparseHostPoolMixin:
    def _round_up_to_page_size(self, size: int) -> int:
        # 将 size 向上对齐到 page_size 的整数倍
        return (size + self.page_size - 1) // self.page_size * self.page_size

    def alloc_page(self, num_pages: int) -> Optional[torch.Tensor]:
        # 按页数分配连续主机缓存
        return self.alloc(num_pages * self.page_size)

    def alloc_paged_token_slots(
        self,
        req_to_host_pool: torch.Tensor,
        req_to_host_pool_allocated_len: torch.Tensor,
        req_pool_idx: int,
        start_pos: int,
        num_tokens: int,
```

```
) -> torch.Tensor:
    """Allocate request host slots by page and return token-granular slots.
       以页为单位分配请求主机槽位，返回 token 粒度的索引。"""
    device = req_to_host_pool.device
    if num_tokens <= 0:
        return torch.empty((0,), dtype=torch.int64, device=device)

    allocated_len = int(req_to_host_pool_allocated_len[req_pool_idx])
    end_pos = start_pos + num_tokens
    page_end = self._round_up_to_page_size(end_pos)
    assert start_pos <= allocated_len

    if page_end > allocated_len:
        num_new_pages = (page_end - allocated_len) // self.page_size
        host_locs = self.alloc_page(num_new_pages)
        if host_locs is None:
            logger.error(
                "HiSparse: host mem pool alloc failed for %d host pages "
                "(req_pool_idx=%d, start_pos=%d, num_tokens=%d)",
                num_new_pages,
                req_pool_idx,
                start_pos,
                num_tokens,
            )
            raise RuntimeError(
                f"HiSparse host mem pool alloc failed for {num_new_pages} pages"
            )

        req_to_host_pool[req_pool_idx, allocated_len:page_end] = host_locs.to(
            device=device, non_blocking=True
        )
        req_to_host_pool_allocated_len[req_pool_idx] = page_end

    return req_to_host_pool[req_pool_idx, start_pos:end_pos]

def allocated_host_indices(
    self,
    req_to_host_pool: torch.Tensor,
    req_pool_idx: int,
    allocated_len: int,
) -> torch.Tensor:
    # 返回已分配且有效的主机索引（过滤掉 -1 填充）
    allocated_len = int(allocated_len)
    host_len = min(
        self._round_up_to_page_size(allocated_len),
        req_to_host_pool.shape[1],
    )
    host_indices = req_to_host_pool[req_pool_idx, :host_len]
    return host_indices[host_indices >= 0]
```

评论区精华

- ShangmingCai 询问 dst 索引是否直接在 hisparse 模块中处理，认为放弃了异质页大小的灵活性，建议未来考虑。作者确认当前统一。
- hzh0425 询问预填充端是否需要更新 send_kvcache_hisparse，结论是该函数被移除。
- xiezhq-hermann 建议将 ensure_host_slots 逻辑移入 memory_pool_host，最终抽象为 HiSparseHostPoolMixin。
- hzh0425 在整体评论中强调需要优化 GPU-CPU 同步，并建议进行 PD 模式基准测试。
- 异质页大小设计权衡 (design): 当前 PR 统一了页大小，未来可通过后处理支持异质。
- P 端是否需要更新 send_kvcache_hisparse (correctness): send_kvcache_hisparse 被移除，P 端直接使用统一接口。
- ensure_host_slots 实现位置 (design): 抽象出 HiSparseHostPoolMixin 并集成到 memory_pool_host。
- GPU-CPU 同步优化和基准测试 (performance): 作者接受意见，待后续优化和测试。

风险与影响

- 风险：
  - 性能风险：引入新的分配路径和 GPU-CPU 同步点，可能在 PD 模式下有性能回归，需要基准测试验证。
  - 兼容性风险：统一页大小后，原有的异质大小布局不再支持，如果未来需要不同设备页大小则需要额外改造。
  - 正确性风险：分配逻辑变更涉及边界条件，如向上取整和 assert，需确保无溢出或越界。
- 影响：
  - 对用户：在启用 HiSparse 时，主机 KV 缓存分配现在以页粒度进行，与设备池一致，提升零拷贝传输效率。
  - 对系统：PD 模式下预填充阶段不再需要特化扩展，传输路径简化。
  - 对团队：代码可维护性提高，主机池分配逻辑集中到 Mixin。
  - 风险标记：核心路径变更，性能回归风险，GPU-CPU 同步增加

关联脉络

- PR #25282 [UnifiedTree] Support deepseek v4 host pool layout: 同样修改了 host pool 的内存布局，与本次 page > 1 功能有承接关系。