PR #30310 完整报告

sgl-project/sglang

Increase the KV cache pool when using indexShare by 15%

合并时间：2026-07-08 11:59

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30310

执行摘要

- 一句话：跳过 DSA topk 层的 indexer KV cache 分配，提升可用 tokens 15%
- 推荐动作：建议重点理解 skip_topk_layers 如何在 KV cache 池构造和操作中实现零大小分配，这是一种避免内存浪费的通用模式。同时注意对未来可能的完全跳过 indexer 初始化的改进方向。

功能与动机

此问题由 @vincentzed 发现。在 skip topk 和 indexer 层上，我们不应分配内存。原实现中每层都分配 index_k_with_scale_buffer，导致大量浪费。通过只对非 topk 层分配，恢复 15% 的 KV cache tokens，每 rank 节省 18 GB。

实现拆解

1. DSATokenToKVPool 构造函数：新增 skip_topk_layers: Optional[List[bool]] 参数，默认为全 False。在分配 index_k_with_scale_buffer 时，对于 skip 层使用 shape = (0, cols) 即零大小张量，避免实际分配内存。
2. 池操作函数的跳过逻辑：在 move_kv_cache、get_cpu_copy、load_cpu_copy、get_state_buf_infos 中，根据 self.skip_topk_layers 跳过对 skip 层的处理，防止索引越界或错误拷贝。
3. 池容量计算修正：在 pool_configurator.py 的 _compute_cell_size 中，仅在非 skip 层计算 indexer 每 token 开销，使 cell_size 准确反映实际内存需求，从而正确计算出更大的 max_total_num_tokens。
4. 模型运行器传递参数：在 model_runner_kv_cache_mixin.py 的 _init_pools 中，非 draft worker 且非 hisparse 模式下，生成 skip_topk_layers 列表并传递给 DSATokenToKVPool。draft worker 和 hisparse 模式仍为所有层分配索引缓冲区。
5. 配套函数引用：新增 dsa_layer_skips_topk 函数（从 model_config 导入），用于查询指定层是否跳过 topk/indexer。

关键文件：

- python/sglang/srt/mem_cache/memory_pool.py（模块 KV 缓存池；类别 source；类型 core-logic；符号 DSATokenToKVPool.init, DSATokenToKVPool.move_kv_cache, DSATokenToKVPool.get_cpu_copy, DSATokenToKVPool.load_cpu_copy）：核心逻辑变更：DSATokenToKVPool 新增 skip_topk_layers 参数，在 __init__中根据该参数分配零大

小张量，并修改 move_kv_cache、get_cpu_copy、load_cpu_copy、get_state_buf_infos 以跳过 skip 层。

- python/sglang/srt/model_executor/pool_configurator.py（模块 池容量计算；类别 source；类型 data-contract；符号 MemoryPoolConfig._compute_cell_size）：池容量计算修正：_compute_cell_size 中根据 dsa_layer_skips_topk 仅对非 skip 层计算 indexer 每 token 开销，使 cell_size 精准反映实际需求。
- python/sglang/srt/model_executor/model_runner_kv_cache_mixin.py（模块 KV 缓存初始化；类别 source；类型 data-contract；符号 ModelRunner._init_pools）：参数传递入口：在 _init_pools 中为非 draft、非 hisparse 的 DSA 模型生成 skip_topk_layers 列表，传递给 DSATokenToKVPool。

关键符号：DSATokenToKVPool.init, DSATokenToKVPool.move_kv_cache, DSATokenToKVPool.get_cpu_copy, DSATokenToKVPool.load_cpu_copy, DSATokenToKVPool.get_state_buf_infos, MemoryPoolConfig._compute_cell_size, ModelRunner._init_pools

关键源码片段

python/sglang/srt/mem_cache/memory_pool.py

核心逻辑变更：DSATokenToKVPool 新增 skip_topk_layers 参数，在 __init__中根据该参数分配零大小张量，并修改 move_kv_cache、get_cpu_copy、load_cpu_copy、get_state_buf_infos 以跳过 skip 层。

```
# python/sglang/srt/mem_cache/memory_pool.py

class DSATokenToKVPool(MLATokenToKVPool):
    # ... 类定义 ...

    def __init__(
        self,
        size: int,
        page_size: int,
        # ... 其他参数 ...
        skip_topk_layers: Optional[List[bool]] = None,
    ):
        # ... 父类初始化 ...
        # 存储层的 skip 标记，未提供时默认全不 skip
        self.skip_topk_layers = (
            list(skip_topk_layers)
            if skip_topk_layers is not None
            else [False] * layer_num
        )
        assert len(self.skip_topk_layers) == layer_num

        # 预计算每页列数，在循环外避免重复计算
        cols = self.page_size * (
            index_head_dim + index_head_dim // self.quant_block_size * 4
```

```
        )
        num_pages = (index_buf_size + page_size + 1) // self.page_size

        self.index_k_with_scale_buffer = [
            torch.zeros(
                # 对 skip 层分配 0 大小，不占用实际内存
                (0 if self.skip_topk_layers[i] else num_pages, cols),
                dtype=self.index_k_with_scale_buffer_dtype,
                device=device,
            )
            for i in range(layer_num) # 遍历每层，按层决定大小
        ]

    def move_kv_cache(self, tgt_loc: torch.Tensor, src_loc: torch.Tensor):
        # Move latent KV and the DSA indexer cache (key + scale) in lockstep.
        super().move_kv_cache(tgt_loc, src_loc)
        if tgt_loc.numel() == 0:
            return
        tgt_loc_flat = tgt_loc.view(-1).long()
        src_loc_flat = src_loc.view(-1).long()
        for i, index_k in enumerate(self.index_k_with_scale_buffer):
            if self.skip_topk_layers[i]:
                continue # skip 层没有有效数据，跳过移动
            index_k[tgt_loc_flat] = index_k[src_loc_flat]
```

评论区精华

本 PR 无实质讨论评论，reviewer Fridge003 直接批准。作者通过多次 rerun e2e 测试（包括 TP、DP 等多种配置）验证准确率不受影响。

- 暂无高价值评论线程

风险与影响

- 风险：核心风险是 skip_topk_layers 的逻辑在 draft worker 和 hisparse 模式下被正确禁用，但若未来新增使用场景未正确配置，可能导致 index buffer 缺失而引发运行时错误。此外，dsa_layer_skips_topk 函数可能随模型配置变化返回错误值。目前测试覆盖仅限于 e2e 测试，缺少单元测试验证 skip 层的正确性。
- 影响：对 DSA 模型用户，内存节省显著（每 rank 18GB），可支持更大的 batch size 或更长序列。功能无变化，精度保持（aime25 91.25%）。对非 DSA 模型无影响。
- 风险标记：配置错误风险，缺少测试覆盖，hisparse/draft 分支逻辑

关联脉络

- 暂无明显关联 PR