PR #31092 完整报告

sgl-project/sglang

Fix post-capture KV sizing for SWA pools

合并时间：2026-07-15 11:06

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31092

执行摘要

- 一句话：修复 SWA pool 未传入 post_capture 标志导致内存分配偏小
- 推荐动作：值得快速合并。属于明确的 bug 修复，逻辑清晰，改动量小。建议阅读 post_capture_kv_sizing_planned 方法中以守卫条件替代复杂 return 语句的设计决策，这是提升可读性和可维护性的好做法。

功能与动机

cctry 在评论中指出 kv_cache_configurator.py 中 SWA pool 的构造缺少 post_capture 参数传递（'the swa plumbing is missing'），导致 post-capture KV sizing 对 hybrid SWA 模型失效。PR body 说明需要将 post_capture_active=self.post_capture_kv_active 传入 SWAKVPool 构造函数。

实现拆解

1. 修改 post_capture_kv_sizing_planned 方法：在 server_args.py 中将 return 语句改为更清晰的守卫条件：先检查通用条件（设备、MLA、fp4_e2m1 等），满足后再通过导入的 is_deepseek_v4 和 is_minimax_sparse 检查模型类型，明确排除尚未实现 post-capture finalization 的 DeepSeek-V4 和 MiniMax sparse 模型。同时新增了 kv_cache_dtype != "fp4_e2m1" 条件以排除 FP4 MHA。
2. 修复 SWA pool 构造：在 kv_cache_configurator.py 的 _build_hybrid_swa_kv_pool 方法中，向 SWAKVPool 构造函数新增 .post_capture_active=self.post_capture_kv_active 参数，确保 hybrid SWA 模型的 KV cache 容量按 post-capture 逻辑计算。

关键文件：

- python/sglang/srt/server_args.py（模块 服务器参数；类别 source；类型 core-logic；符号 post_capture_kv_sizing_planned）：修改了 post_capture_kv_sizing_planned 方法，重构了返回值逻辑并增加了 FP4、DeepSeek-V4、MiniMax sparse 的排除条件。
- python/sglang/srt/mem_cache/kv_cache_configurator.py（模块 缓存配置；类别 source；类型 core-logic；符号 _build_hybrid_swa_kv_pool）：在 _build_hybrid_swa_kv_pool 中补上了 post_capture_active=self.post_capture_kv_active 参数，这是核心修复。

关键符号：post_capture_kv_sizing_planned, _build_hybrid_swa_kv_pool

关键源码片段

python/sglang/srt/server_args.py

修改了 post_capture_kv_sizing_planned 方法，重构了返回值逻辑并增加了 FP4、DeepSeek-V4、MiniMax sparse 的排除条件。

```
# 文件：python/sglang/srt/server_args.py
# 函数：ServerArgs.post_capture_kv_sizing_planned
def post_capture_kv_sizing_planned(self) -> bool:
    """是否计划使用 post-capture KV sizing；
    若返回 False，则后备使用保守静态内存比例。
    排除尚未实现 post-capture finalization 的池类型。"""
    use_mla = self.use_mla_backend
    # 守卫条件：逐个排除不支持的场景
    if not (
        envs.SGLANG_ENABLE_POST_CAPTURE_KV_SIZING.get()
        and self.device == "cuda"
        and self.dcp_size == 1
        and not (use_mla() if callable(use_mla) else use_mla)
        and self.kv_cache_dtype != "fp4_e2m1" # FP4 MHA 不支持
        and not self.prefill_only_disable_kv_cache
        and not self.enable_memory_saver
        and envs.SGLANG_MOONCAKE_CUSTOM_MEM_POOL.get() is None
        and not self.enable_dp_attention
        and (
            self.disaggregation_mode == "decode"
            or self.cuda_graph_config.prefill.backend != Backend.DISABLED
        )
        and (
            self.disaggregation_mode == "prefill"
            or self.cuda_graph_config.decode.backend != Backend.DISABLED
        )
    ):
        return False

    # 运行时检查模型类型：排除 DeepSeek-V4 和 MiniMax sparse
    from sglang.srt.configs.model_config import is_deepseek_v4, is_minimax_sparse
    hf_config = self.get_model_config().hf_config
    return not (is_deepseek_v4(hf_config) or is_minimax_sparse(hf_config))
```

python/sglang/srt/mem_cache/kv_cache_configurator.py

在 _build_hybrid_swa_kv_pool 中补上了 post_capture_active=self.post_capture_kv_active 参数，这是核心修复。

```
# 文件：python/sglang/srt/mem_cache/kv_cache_configurator.py
# 方法：KVCacheConfigurator._build_hybrid_swa_kv_pool
def _build_hybrid_swa_kv_pool(
    self,
    *,
    full_max_total_num_tokens: Optional[int],
    swa_max_total_num_tokens: Optional[int],
```

```
        mha_pool_class: type,
    ) -> KVCache:
        kwargs = {}
        if self.is_hybrid_swa_compress:
            kwargs = {
                "swa_head_num": max(
                    1,
                    self.model_config.hf_text_config.swa_num_key_value_heads
                    // get_parallel().attn_tp_size,
                ),
                "swa_head_dim": self.model_config.swa_head_dim,
                "swa_v_head_dim": self.model_config.swa_v_head_dim,
                "v_head_dim": self.model_config.v_head_dim,
            }
        token_to_kv_pool = SWAKVPool(
            size=full_max_total_num_tokens,
            size_swa=swa_max_total_num_tokens,
            page_size=self.server_args.page_size,
            dtype=self.kv_cache_dtype,
            post_capture_active=self.post_capture_kv_active, # 新增：传递 post-capture 标志
            head_num=self.model_config.get_num_kv_heads(get_parallel().attn_tp_size),
            head_dim=self.model_config.head_dim,
            swa_attention_layer_ids=self.model_config.swa_attention_layer_ids,
            full_attention_layer_ids=self.model_config.full_attention_layer_ids,
            device=self.device,
            enable_kv_cache_copy=(self.server_args.speculative_algorithm is not None),
            token_to_kv_pool_class=mha_pool_class,
            **kwargs,
        )
        return token_to_kv_pool
```

评论区精华

未找到 review 讨论。合并者 merrymercy 直接修复了 cctry 指出的遗漏。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。修改小巧（+11/-3），核心逻辑是补全参数传递和增加排除条件。风险点在于：新增的 is_deepseek_v4 和 is_minimax_sparse 导入可能影响 post_capture_kv_sizing_planned 方法在模型配置未就绪时的行为（但该方法已在 get_model_config 之后调用）。此外，FP4 排除条件 kv_cache_dtype != "fp4_e2m1" 可能与其他 dtype 字符串格式不一致（如 "fp4"），但当前确认使用 "fp4_e2m1"。
- 影响：影响范围：启用 post-capture KV sizing 且使用 hybrid SWA 模型（如某些长上下文模型）的用户。这些用户的 SWA KV pool 容量先前被低估，可能导致 OOM 或性能下降；修复后内存分配更准确。对于非 SWA 模型无影响。影响程度较小。
- 风险标记：低风险，改动小

关联脉络

- PR #31173 [PD] Stride KV token->page indices on device before D2H copy: 同为 KV cache 相关优化，涉及 post-capture KV sizing 的上下游逻辑。
- PR #30351 [Bug fix] Account for KV replication fan-out in transfer-byte metrics: 同为 KV cache 指标与内存管理相关的 bugfix。