PR #25022 完整报告

sgl-project/sglang

[Bugfix, NSA HiCache] Fix missing override_kv_cache_dim in attach_hybrid_nsa_pool_to_hiradix_cache

合并时间：2026-05-13 11:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25022

执行摘要

- 一句话：修复 NSA HiCache 中缺失的 override_kv_cache_dim 参数
- 推荐动作：值得精读，展示如何从重构中引入细微 bug 并修复，同时进行接口清理。关注 override_kv_cache_dim 在共享锚点中的必要性，以及改为传递通信组对象的设计思路。

功能与动机

根据 PR body，由于在重构 hybrid_pool_assembler.py（PR#23243）时遗漏参数，导致运行时出现 RuntimeError: The size of tensor a (576) must match the size of tensor b (656) at non-singleton dimension 2。修复此问题以恢复 NSA/DeepSeek 模型在 HiCache 下的正常推理。

实现拆解

1. 在 attach_hybrid_nsa_pool_to_hiradix_cache 函数中，向 build_shared_anchor_stack 调用添加 override_kv_cache_dim=kv.kv_cache_dim 参数。
2. 在 build_kv_only_stack、build_hybrid_swa_stack、build_hybrid_mamba_stack、build_shared_anchor_stack 中移除不再需要的 attn_cp_rank 和 attn_cp_size 参数。
3. 在上述函数中新增 attn_cp_group 和 attn_tp_group 参数，并在构造 HybridCacheController 时传递它们，替换原来的 rank/size 参数。
4. 在 unified_radix_cache.py 的 init_hicache 方法中，调用 attach_hybrid_pool_to_unified_cache 时传递 attn_cp_group=params.attn_cp_cache_group 和 attn_tp_group=params.attn_tp_cache_group，确保通信组信息正确传播。
5. hybrid_cache_controller.py 中移除 __init__ 的 attn_cp_rank 和 attn_cp_size 参数，这些信息已由通信组对象直接携带。未直接添加测试文件，改动依赖回归测试。

关键文件：

- python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py（模块 缓存层；类别 source；类型 core-logic；符号 attach_hybrid_nsa_pool_to_hiradix_cache, build_kv_only_stack, build_hybrid_swa_stack, build_hybrid_mamba_stack）：核心修复：在 attach_hybrid_nsa_pool_to_hiradix_cache 中添加缺失的 override_kv_cache_dim 参数；同时清理多个 build_*_stack 函数的参数，使用 attn_cp_group/attn_tp_group 替代 attn_cp_rank/attn_cp_size。

- python/sglang/srt/mem_cache/hybrid_cache/hybrid_cache_controller.py（模块 缓存层；类别 source；类型 entrypoint；符号 HybridCacheController.init）：配合参数清理，移除 HybridCacheController.__init__中的 attn_cp_rank 和 attn_cp_size 参数。
- python/sglang/srt/mem_cache/unified_radix_cache.py（模块 缓存层；类别 source；类型 core-logic；符号 UnifiedRadixCache.init_hicache）：在 init_hicache 中调用 attach_hybrid_pool_to_unified_cache 时传递 attn_cp_group 和 attn_tp_group，确保通信组信息传递给下级构造。

关键符号：attach_hybrid_nsa_pool_to_hiradix_cache, build_kv_only_stack, build_hybrid_swa_stack, build_hybrid_mamba_stack, build_shared_anchor_stack, HybridCacheController.init, UnifiedRadixCache.init_hicache

关键源码片段

python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py

核心修复：在 attach_hybrid_nsa_pool_to_hiradix_cache 中添加缺失的 override_kv_cache_dim 参数；同时清理多个 build_*_stack 函数的参数，使用 attn_cp_group/attn_tp_group 替代 attn_cp_rank/attn_cp_size。

```
def build_kv_only_stack(
    *,
    params: CacheInitParams,
    server_args: ServerArgs,
    kv_pool: Any,
    full_layer_mapping: dict[int, int],
    page_size: int,
    tp_group,
    load_cache_event,
    # 新增参数：替换了旧的 attn_cp_rank 和 attn_cp_size
    attn_cp_group: Optional["torch.distributed.ProcessGroup"] = None,
    attn_tp_group: Optional["torch.distributed.ProcessGroup"] = None,
    storage_backend: Optional[str],
    use_mla: bool,
    override_kv_cache_dim: Optional[int] = None, # 可覆盖 kv 缓存维度
    prefetch_threshold: int = 256,
    model_name: Optional[str] = None,
    storage_backend_extra_config: Optional[dict] = None,
    pp_rank: int = 0,
    pp_size: int = 1,
    enable_storage_metrics: bool = False,
) -> tuple[HostPoolGroup, HybridCacheController]:
    transfer_layer_num = len(full_layer_mapping)
    kv_host_pool = build_kv_host_pool(
        kv_pool=kv_pool, page_size=page_size,
        server_args=server_args, use_mla=use_mla,
        override_kv_cache_dim=override_kv_cache_dim,
    )
    entries = [
```

```
            build_pool_entry(
                name=PoolName.KV, host_pool=kv_host_pool,
                device_pool=kv_pool, layer_mapping=full_layer_mapping,
                transfer_layer_num=transfer_layer_num, is_anchor=True,
            )
        ]
        host_pool_group = HostPoolGroup(entries)
        cache_controller = HybridCacheController(
            params.token_to_kv_pool_allocator, host_pool_group,
            page_size, tp_group,
            load_cache_event=load_cache_event,
            attn_cp_group=attn_cp_group, # 传递通信组对象而非 rank
            attn_tp_group=attn_tp_group,
            write_policy=server_args.hicache_write_policy,
            io_backend=server_args.hicache_io_backend,
            storage_backend=storage_backend,
            prefetch_threshold=prefetch_threshold,
            model_name=model_name,
            storage_backend_extra_config=storage_backend_extra_config,
            pp_rank=pp_rank, pp_size=pp_size,
            transfer_layer_num=transfer_layer_num,
            enable_storage_metrics=enable_storage_metrics,
        )
        return host_pool_group, cache_controller
```

python/sglang/srt/mem_cache/unified_radix_cache.py

在 init_hicache 中调用 attach_hybrid_pool_to_unified_cache 时传递 attn_cp_group 和 attn_tp_group，确保通信组信息传递给下级构造。

```
    def init_hicache(self, server_args: ServerArgs, params: CacheInitParams) -> None:
        """Initialize HiCache infrastructure."""
        from sglang.srt.mem_cache.hybrid_cache.hybrid_pool_assembler import (
            attach_hybrid_pool_to_unified_cache,
        )

        # Direct IO layout fixup (must happen before pool creation)
        if server_args.hicache_io_backend == "direct":
            if server_args.hicache_mem_layout == "page_first":
                server_args.hicache_mem_layout = "page_first_direct"
                logger.warning(
                    "Page first layout is not supported with direct IO backend, "
                    "switching to page first direct layout"
                )

        self.load_cache_event = threading.Event()
        self.hicache_anchor_kv_shared_indices_pools.clear()
        attach_hybrid_pool_to_unified_cache(
            self,
            params,
```

```
        server_args,
        load_cache_event=self.load_cache_event,
        # 新增传递通信组参数，替换之前的 attn_cp_rank/attn_cp_size
        attn_cp_group=params.attn_cp_cache_group,
        attn_tp_group=params.attn_tp_cache_group,
    )

    # State initialization
    self.write_through_threshold = (
        1 if server_args.hicache_write_policy == "write_through" else 2
    )
    self.load_back_threshold = 256

    logger.info(
        f"HiCache D\u2194H initialized: "
        f"host_pool_size={self.host_pool_group.size}, "
        f"write_policy={server_args.hicache_write_policy}, "
        f"tp_world_size={self.tp_world_size}, "
        f"transfer_layer_num={self.cache_controller.layer_num}"
    )
```

评论区精华

无实质性讨论，PR 由 hzh0425 审批通过，未提出额外问题。

- 暂无高价值评论线程

风险与影响

- 风险：核心修复仅涉及一行参数添加，改动明显，回归风险低。参数清理可能影响其他未发现的调用方式，但所有修改点在代码中一致更新。主要风险是缺少针对 NSA+HiCache 的组合测试覆盖，若未来有类似重构容易再次遗漏。
- 影响：影响所有使用 NSA/DeepSeek 模型并启用 HiCache 的用户，修复了运行时崩溃。参数清理统一了通信组传递方式，简化了接口，对开发者友好。影响范围集中在 HiCache 路径，不涉及其他功能。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #23243 [Hybrid-Cache]: Refactor hybrid_pool_assembler.py: 该重构引入了 missing override_kv_cache_dim 的 bug，本 PR 为 bugfix。