PR #24040 完整报告

sgl-project/sglang

[CP] Register KV cache allgather buffer with symmetric memory

合并时间：2026-05-06 23:24

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24040

执行摘要

- 一句话：修复 cp_all_gather 缓冲区未注册对称内存
- 推荐动作：值得快速合并。这是一个明确的遗漏修复，逻辑简单，风险低。开发者可关注 #22914 重构后的对称内存接口一致性；reviewer 可参考讨论中关于预分配缓冲池的策略。

功能与动机

PR #22914 将 NSA 的 cp 工具函数合并到 cp_utils.py 后，遗漏了在 cp_all_gather_reorganized_into_tensor_kv_cache 中使用 use_symmetric_memory 包裹缓冲区分配。这导致对称内存能力被遗漏，影响通信效率。PR body 明确说明这是 #22914 的 follow-up，原始对称内存支持来自 #17756。

实现拆解

1. 在 python/sglang/srt/layers/utils/cp_utils.py 的 cp_all_gather_reorganized_into_tensor_kv_cache 函数中，将创建 input_tensor_full 的 torch.empty 调用包裹在 use_symmetric_memory 上下文管理器内。
2. 通过 is_allocation_symmetric() 动态判断是否启用对称内存（disabled=not is_allocation_symmetric()），实现兼容性。
3. 其他逻辑不变（padding、split、cat 等），确保行为一致。
4. 无测试配套变更，但后续 CI 验证已通过。

关键文件：

- python/sglang/srt/layers/utils/cp_utils.py（模块 cp 工具；类别 source；类型 core-logic；符号 cp_all_gather_reorganized_into_tensor_kv_cache, use_symmetric_memory, is_allocation_symmetric）：核心变更文件，修复了 context parallel allgather 缓冲区对称内存注册的遗漏。

关键符号：cp_all_gather_reorganized_into_tensor_kv_cache

关键源码片段

python/sglang/srt/layers/utils/cp_utils.py

核心变更文件，修复了 context parallel allgather 缓冲区对称内存注册的遗漏。

```
# 路径：python/sglang/srt/layers/utils/cp_utils.py
```

```
# 函数：cp_all_gather_reorganized_into_tensor_kv_cache
# 上下文：在创建 allgather 输出缓冲区时，使用对称内存上下文管理器

def cp_all_gather_reorganized_into_tensor_kv_cache(
    input_tensor, total_len, cp_size, forward_batch, stream
):
    """
    Allgather communication for context_parallel KV cache.
    Handles multi-dimensional tensors (e.g., [seq_len, num_heads, head_dim]).
    """
    max_len = (total_len + cp_size - 1) // cp_size
    pad_size = max_len - input_tensor.shape[0]
    if pad_size > 0:
        # 填充第一维 (seq_len)，F.pad 需要按维度逆序传入填充参数
        padding = [0, 0] * (input_tensor.ndim - 1) + [0, pad_size]
        input_tensor = F.pad(input_tensor, padding, mode="constant", value=0)

    # 使用对称内存上下文创建输出缓冲区，以注册对称内存提升通信效率
    # is_allocation_symmetric() 返回 False 时回退到普通分配
    with use_symmetric_memory(
        get_attention_cp_group(), disabled=not is_allocation_symmetric()
    ):
        input_tensor_full = torch.empty(
            max_len * cp_size,
            *input_tensor.shape[1:],
            device=input_tensor.device,
            dtype=input_tensor.dtype,
        )

    get_attention_cp_group().cp_all_gather_into_tensor_async(
        input_tensor_full, input_tensor, stream
    )
    # ... 后续 split 和 cat 逻辑不变
```

评论区精华

reviewer Shunkangz 询问动态长度场景下是否会在运行时通过 ncclMemAlloc 分配（若池大小不足），作者 wangfakang 确认有此可能，但指出启动时默认预分配 4GB 对称内存缓冲池以减少运行时分配。最终结论是当前方案合理，无未解决疑虑。

- 运行时内存分配与预热策略 (question): wangfakang 确认存在运行时分配的可能，但指出启动时默认预分配 4GB 池来尽量避免频繁分配。

风险与影响

- 风险：风险低。变更仅将 torch.empty 调用包裹在上下文管理中，不改变函数签名或逻辑流。若 is_allocation_symmetric() 返回 False（对称内存不可用），行为完全回退到原样。可能的风险是依赖 get_attention_cp_group() 返回的有效组，但原函数已依赖该组执行

allgather，因此无新增依赖风险。

- 影响：影响范围小，仅影响启用了对称内存的 context parallel 场景。对于未启用对称内存的环境，无影响。对于启用的环境，KV cache allgather 缓冲区现在会通过对称内存注册，减少不必要的通信拷贝，提升效率。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #22914 [Refactor] Deduplicate NSA utils.py into cp_utils.py for context parallel: 本 PR 直接修复 #22914 重构时遗漏的对称内存注册。
- PR #17756 Register cp-atten-allgather buffers with symm memory: 原始对称内存注册 PR，本 PR 是将相同能力补回到新函数中。