PR #31688 完整报告

sgl-project/sglang

Fix ROCm fused KV and KDA paths

合并时间：2026-07-20 06:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31688

执行摘要

- 一句话：修复 ROCm 融合 KV 和 KDA 路径的同步与池类型问题
- 推荐动作：值得精读的部分在于处理统一池与混合池差异的设计决策，以及 Triton 内核中同步屏障的添加方式。对于维护 ROCm 后端的工程师有直接参考价值。

功能与动机

ROCm 后端在 fused KV 缓存和 KDA 路径上存在数据一致性问题：KDA 内核中跨 warp 重载 tile 缺少同步屏障；融合 KV 缓存设置中未区分统一池和混合池，在非混合池（--disable-hybrid-swa-memory）下访问了不存在的 full_to_swa_index_mapping，导致错误。

实现拆解

1. python/sglang/kernels/ops/attention/fla/chunk_intra.py: 在 chunk_kda_fwd_kernel_inter_solve_fused 内核中，于全局 tile 存储（tl.store）之后、跨 warp 进行 forward substitution 重载之前，插入 tl.debug_barrier()，确保所有 warp 的数据可见性。
2. python/sglang/srt/models/utils.py: 在 create_fused_set_kv_buffer_arg 函数的 ROCm 分支中，先通过 isinstance(token_to_kv_pool, SWAKVPool) 判断池类型，再决定是否获取 full_to_swa_index_mapping；仅当池为混合池且层有滑动窗口时，才设置 slot_mapping_swa，否则设为 None。同时将 FusedSetKVBufferArg 的导入移到文件级，并添加注释说明 CUDA/ROCm 的差异。

关键文件：

- python/sglang/srt/models/utils.py（模块 模型工具；类别 source；类型 data-contract；符号 create_fused_set_kv_buffer_arg, enable_fused_set_kv_buffer）：核心修复文件：重写 ROCm 分支中 fused KV 缓存参数的构建逻辑，区分统一池和混合池，并调整导入方式。
- python/sglang/kernels/ops/attention/fla/chunk_intra.py（模块 注意力内核；类别 infra；类型 infrastructure；符号 chunk_kda_fwd_kernel_inter_solve_fused）：在内核内添加 tl.debug_barrier()，解决 KDA fused store 后跨 warp 重载的同步问题。

关键符号：create_fused_set_kv_buffer_arg, chunk_kda_fwd_kernel_inter_solve_fused

关键源码片段

python/sglang/srt/models/utils.py

核心修复文件：重写 ROCm 分支中 fused KV 缓存参数的构建逻辑，区分统一池和混合池，并调整导入方式。

```
def create_fused_set_kv_buffer_arg(
    value: torch.Tensor,
    layer: RadixAttention,
    forward_batch: ForwardBatch,
):
    layer_id = layer.layer_id
    token_to_kv_pool = get_token_to_kv_pool()

    k_buffer = token_to_kv_pool.get_key_buffer(layer_id)
    v_buffer = token_to_kv_pool.get_value_buffer(layer_id)

    if not _is_hip:
        # CUDA path: only bf16 KV cache, no scaling support.
        assert layer.k_scale is None and layer.v_scale is None, "scale not supported"
        return FusedSetKVBufferArg(
            value=value,
            k_buffer=k_buffer.view(k_buffer.shape[0], -1),
            v_buffer=v_buffer.view(v_buffer.shape[0], -1),
            cache_loc=forward_batch.out_cache_loc,
        )
    else:
        # ROCm path: support bf16/fp16/fp8 KV cache.
        page_size = token_to_kv_pool.page_size
        # A non-hybrid pool has no full->SWA remap: SWA and full layers
        # share one slot space indexed directly by out_cache_loc (as
        # --disable-hybrid-swa-memory gives). Leaving swa_slot_mapping=None
        # makes the fused store write at out_cache_loc, matching CUDA path.
        full_to_swa = (
            token_to_kv_pool.full_to_swa_index_mapping
            if isinstance(token_to_kv_pool, SWAKVPool)
            else None
        )
        slot_mapping_swa = (
            full_to_swa.long()
            if layer.sliding_window_size > 0 and full_to_swa is not None
            else None
        )
        # ... rest of the function uses slot_mapping_swa ...
```

python/sglang/kernels/ops/attention/fla/chunk_intra.py

在内核内添加 tl.debug_barrier()，解决 KDA fused store 后跨 warp 重载的同步问题。

```
@triton.jit
def chunk_kda_fwd_kernel_inter_solve_fused(...):
    # ... kernel computation ...
```

```
    # Store results to global memory
    tl.store(p_Akkd11, b_Akk_d1.to(Akkd.dtype.element_ty), boundary_check=(0, 1))
    tl.store(p_Akkd22, b_Akk_d2.to(Akkd.dtype.element_ty), boundary_check=(0, 1))
    tl.store(p_Akkd33, b_Akk_d3.to(Akkd.dtype.element_ty), boundary_check=(0, 1))
    # Forward substitution reloads these global tiles across warps below.
    tl.debug_barrier()
    b_Ai00 = b_Akk_d0
    b_Ai11 = b_Akk_d1
    # ... continue with forward substitution ...
```

评论区精华

GeminiCodeAssist的自动审查未提出具体问题，仅概述了变更内容，并提示该工具即将停用。无人工 review 评论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：KDA 屏障仅影响 Triton kernel 内跨 warp 同步，语义安全；池类型检查通过 isinstance 显式判断，避免动态属性访问的隐式错误。未引入新依赖或核心逻辑变更。
- 影响：影响范围窄：仅修复 ROCm 后端的两个特定路径。用户层面，AMD GPU 用户可能在某些配置下（如 --disable-hybrid-swa-memory 或 KDA 注意力）遇到之前的错误，本 PR 修正后不再发生。系统层面，非 ROCm 后端不受影响。
- 风险标记：低风险，仅影响 ROCm 后端

关联脉络

- PR #31687 [Scheduler] Move the WAR barrier to right after each run_batch launch: 同为同步屏障的修复，但位于调度器层级，讨论 WAR barrier 的正确放置。
- PR #29353 [Scheduler] Add SGLANG_FORCE_COARSE_WAR_BARRIER opt-in for a whole-forward WAR barrier: 引入粗粒度 WAR 屏障环境变量，与 KDA 内核的内联屏障属于同一类问题（跨 warp/stream 同步）。
- PR #31474 Fix KDA prefix caching under mamba extra_buffer and enable it for kimi_linear: 之前对 KDA 后端的修复，本 PR 在此基础上补充了 ROCm 特定同步修复。