PR #29267 完整报告

sgl-project/sglang

[CPU] add indices in chunk_gated_delta_rule

合并时间：2026-06-26 07:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29267

执行摘要

- 一句话：CPU GDN kernel 内联状态索引，移除外部 gather
- 推荐动作：值得精读。该 PR 展示了如何将常见的外部 gather/scatter 操作通过 kernel 参数下沉到计算内部，减少 host-device 同步和数据搬运。对于优化 CPU/GPU 推理后端有参考价值。重点关注 fla.cpp 中索引修改和 gdn_triton.py 中调用侧的适配。

功能与动机

PR body 明确指出 'adds indices support in chunk_gated_delta_rule, so as to remove index and index_put before and after'，并附带了性能对比图（img）。该变更旨在减少外部索引操作，将状态选择内联到 kernel 中，提升整体效率。

实现拆解

1. CPU C++ kernel 添加 indices 参数：在 sgl-kernel/csrc/cpu/mamba/fla.cpp 中，chunk_gated_delta_rule_fwd_inter_kernel_impl 增加 indices 参数，state 指针计算由 state + bs * stride 改为 state + indices[bs] * stride，直接索引状态池。同时上游调用函数 (chunk_gated_delta_rule_fwd_inter, chunk_gated_delta_rule_cpu) 传递该参数。
2. Python 层统一传递 indices：在 gdn_triton.py 的 extend 方法中，移除对 CPU 的特殊处理（不再用 ssm_states[cache_indices] 预先 gather），改为始终传递 initial_state_indices 参数，让 CPU kernel 内部索引。NPU 仍保持外部 gather 方式。
3. CPU graph runner fake 适配：cpu_graph_runner.py 中 chunk_gated_delta_rule_cpu 的 fake 签名添加 initial_state_indices 参数。
4. 门控融合回退路径修补：qwen3_5.py 中将 if not _is_npu: 改为 if not (_is_npu or _is_cpu):，使 CPU 也不走 Triton fused_sigmoid_mul，而是使用 torch 原生 mul_ + sigmoid，避免不兼容的 kernel。
5. Torch 注册和 Python wrapper 更新：torch_extension_cpu.cpp 更新 op schema 和函数声明；mamba.py 更新 Python 包装函数添加 initial_state_indices 参数。
6. 单元测试增强：test_mamba.py 的 test_chunk_gated_delta_rule 使用 cache_indices 模拟索引状态池，验证返回 returned_state 与初始状态一致，并确认未使用的槽位未被修改。

关键文件：

- sgl-kernel/csrc/cpu/mamba/fla.cpp（模块 CPU 内核；类别 source；类型 core-logic；符号 chunk_gated_delta_rule_fwd_inter_kernel_impl, chunk_gated_delta_rule_fwd_inter,

chunk_gated_delta_rule_cpu）：核心 CPU kernel 文件，实现状态索引下沉，修改 state 指针计算方式，是性能优化的关键。

- python/sglang/srt/layers/attention/linear/kernels/gdn_triton.py（模块 注意力内核；类别 source；类型 core-logic；符号 extend）：Python 侧调用入口，移除对 CPU 的特殊 gather 逻辑，统一传递 indices 参数。
- test/registered/cpu/test_mamba.py（模块 测试；类别 test；类型 test-coverage；符号 TestMambaAttention.test_chunk_gated_delta_rule）：新增覆盖索引状态池行为的单元测试，验证 kernel 内部索引的正确性。

关键符号：chunk_gated_delta_rule_fwd_inter_kernel_impl, chunk_gated_delta_rule_fwd_inter, chunk_gated_delta_rule_cpu, extend, self_attention

关键源码片段

sgl-kernel/csrc/cpu/mamba/fla.cpp

核心 CPU kernel 文件，实现状态索引下沉，修改 state 指针计算方式，是性能优化的关键。

```
// sgl-kernel/csrc/cpu/mamba/fla.cpp
// 函数签名新增 indices 参数，用于直接索引状态池
template <typename scalar_t, int D, int CHUNK_SIZE>
void chunk_gated_delta_rule_fwd_inter_kernel_impl(
    scalar_t* __restrict__ out,
    float* __restrict__ state,
    const int32_t* __restrict__ indices, // <-- 新增：状态池索引
    const scalar_t* __restrict__ q,
    ...
) {
  // 并行遍历 batch * head
  at::parallel_for(0, num_seqs * Hv, 0, [&](int64_t begin, int64_t end) {
    ...
    // 步骤 2.a: 使用 indices[bs] 替代 bs 直接索引 state 槽位
    float* __restrict__ s_ptr = state + indices[bs] * (Hv * D * D) + hv * (D * D);
    // 其余计算相同：将 state 乘以 exp(g_last)，执行 brgemm 等
    ...
  });
}

// 上层封装传递 initial_state_indices
std::tuple<at::Tensor, at::Tensor> chunk_gated_delta_rule_cpu(
    ...
    const at::Tensor& initial_state_indices,
    double eps) {
  // num_seqs 从 initial_state_indices 的 size 0 获取
  const int64_t num_seqs = initial_state_indices.size(0);
  // 将索引传入 kernel 实现
  chunk_gated_delta_rule_fwd_inter_kernel_impl<...>(
      o.data_ptr<scalar_t>(),
      initial_state.data_ptr<float>(),
```

```
        initial_state_indices.data_ptr<int32_t>(),
        ...);
}
```

python/sglang/srt/layers/attention/linear/kernels/gdn_triton.py

Python 侧调用入口，移除对 CPU 的特殊 gather 逻辑，统一传递 indices 参数。

```
# python/sglang/srt/layers/attention/linear/kernels/gdn_triton.py
def extend(self, ..., *, ssm_states, cache_indices, query_start_loc, **kwargs):
    recurrent_state = ssm_states
    # 构造包含 indices 的参数字典
    recurrent_state_indices_args = {"initial_state_indices": cache_indices}
    # 仅在 NPU 上仍保持外部 gather，CPU 则不再需要
    if is_npu():
        recurrent_state = ssm_states[cache_indices]
        recurrent_state_indices_args = {}
    # 统一调用 kernel，CPU 将 indices 传入 kernel 内部处理
    return chunk_gated_delta_rule(
        q=q, k=k, v=v, g=g, beta=beta,
        initial_state=recurrent_state,
        cu_seqlens=query_start_loc,
        head_first=False,
        use_qk_l2norm_in_kernel=True,
        **recurrent_state_indices_args, # CPU 传入 indices，NPU 为空
    )
```

评论区精华

该 PR 未收到公开 review 评论，无讨论线程。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. kernel 内部索引越界风险： indices[bs] 的值直接用于状态池偏移，若 indices 不在 [0, pool_size) 内将导致内存访问违规。依赖调用方保证 indices 合法。
  2. 精度回归风险：状态计算路径发生改变（由外部 gather 改为内部索引），对浮点计算顺序可能产生微小差异，但测试未发现精度问题。
  3. NPU/GPU 无影响：变更仅影响 CPU 路径，其他硬件后端逻辑保持不变。
  4. Triton 门控回退路径变更：将 CPU 从 fused_sigmoid_mul Triton kernel 切换到 torch 回退，可能改变浮点舍入行为，但门控是逐元素操作，影响极小。 - 影响：用户影响：CPU 推理用户可直接受益于性能提升（减少数据搬运和 kernel launch 开销），不改变 API。系统影响：kernel 内部直接索引减少了外部临时 tensor 创建，降低内存带宽压力。团队影响：为后续其他后端（如 GPU）将索引下沉到 kernel 提供了参考模式。
- 风险标记：核心路径变更（CPU kernel 索引逻辑），CPU-only kernel 变更，测试覆盖增加后仍有边界风险

关联脉络

- 暂无明显关联 PR