PR #25514 完整报告

sgl-project/sglang

[diffusion] Clean up VSA attention hot path

合并时间：2026-05-24 16:46

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25514

执行摘要

- 一句话：优化 VSA 注意力热点路径，复用 tile buffer 并预计算 untile 索引
- 推荐动作：值得精读 tile buffer 复用和预计算索引的设计模式，可推广至其他需要频繁分配临时缓冲区的热点路径。denoising 中优先选择 sparse backend 的决策也值得关注。但对于新增参数 reviewer 意见未采纳，需关注后续是否带来兼容性成本。

功能与动机

来自 FastVideo PR #1272，需要清理 VSA 注意力热点路径，减少不必要的显存分配和 kernel 启动，提升 diffusion 模型推理性能。同时修复 Wan 模型在 VSA 路径下的兼容性问题，使 --attention-backend video_sparse_attn 配置可正常使用。

实现拆解

1. VideoSparseAttentionMetadata 扩展：新增 untile_combined_index: torch.LongTensor 和 tile_buf: torch.Tensor | None 字段，前者在 build() 中通过 non_pad_index[reverse_tile_partition_indices] 预计算，后者初始为 None，用于缓存 padded 缓冲区。
2. tile 方法改造：改为接收 attn_metadata 而非分散参数，从 metadata 读取 tile_buf 并检查形状 / 类型 / 设备，匹配则复用，否则重新分配并更新 metadata。
3. untile 方法简化：由两次 fancy index（x[:, non_pad_index][:, reverse_tile_partition_indices]）改为单次索引 x[:, untile_combined_index]，减少 kernel 启动。
4. preprocess_qkv / postprocess_output 简化：直接调用改造后的 tile/untile，传递 metadata 而非多个参数。
5. SparseLinearAttention 去除冗余 .contiguous()：feature_map_q/k 输出已满足 layout 要求，无需额外 contiguous 调用。
6. DenoisingStage 改进：_infer_transformer_attention_backend 在多个 backend 时优先选择 is_sparse 的 backend；_build_attn_metadata 中 VSA_sparsity 读取支持 sparsity 作为备用键。
7. Wan 模型参数添加：在 WanTransformerBlock.__init__ 中新增 attention_type 和 sla_topk 参数，允许 VSA 注意力后端使用通用 Wan block kwargs 和混合 FA/VSA 元数据。

关键文件：

- python/sglang/multimodal_gen/runtime/layers/attention/backends/video_sparse_attn.py（模块 VSA 后端；类别 source；类型 core-logic）：核心修改文件：metadata 新增字段实现 tile buffer 复用和 untile 索引预计算，tile/untile 方法改造为接收 metadata 并复用缓冲区。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py（模块 去噪阶段；类别 source；类型 core-logic）：修改 denoising 阶段的 attention backend 推断逻辑和配置读取，修复多 backend 混用时的选择优先级并兼容 VSA_sparsity 键名。
- python/sglang/multimodal_gen/runtime/layers/attention/backends/sparse_linear_attn.py（模块 稀疏线性注意力；类别 source；类型 core-logic）：移除 feature map 后冗余的 .contiguous() 调用，减少不必要的内存操作。
- python/sglang/multimodal_gen/runtime/models/dits/wanvideo.py（模块 Wan 模型；类别 source；类型 data-contract）：添加 attention_type 和 sla_topk 参数以支持 VSA 路径兼容 Wan 模型，允许 VSA 注意力后端使用通用 Wan block kwargs 和混合 FA/VSA 元数据。
- python/sglang/multimodal_gen/test/unit/test_video_sparse_attention.py（模块 VSA 测试；类别 test；类型 test-coverage；符号 test_video_sparse_attention_tile_buffer_reuse_and_untile）：新增 VSA tile buffer 重用与 untile 正确性单元测试，验证缓存复用和组合索引正确性。

关键符号：VideoSparseAttentionImpl.tile, VideoSparseAttentionImpl.untile, VideoSparseAttentionImpl.preprocess_qkv, VideoSparseAttentionImpl.postprocess_output, DenoisingStage._infer_transformer_attention_backend, DenoisingStage._build_attn_metadata, WanTransformerBlock.init, SparseLinearAttention.forward

关键源码片段

python/sglang/multimodal_gen/runtime/layers/attention/backends/video_sparse_attn.py

核心修改文件：metadata 新增字段实现 tile buffer 复用和 untile 索引预计算，tile/untile 方法改造为接收 metadata 并复用缓冲区。

```
# 关键变更：tile 方法改为复用 attn_metadata.tile_buf，避免每次分配新 buffer
def tile(
    self,
    x: torch.Tensor,
    attn_metadata: VideoSparseAttentionMetadata,
) -> torch.Tensor:
    num_tiles = attn_metadata.num_tiles
    t_padded_size = num_tiles[0] * VSA_TILE_SIZE[0]
    h_padded_size = num_tiles[1] * VSA_TILE_SIZE[1]
    w_padded_size = num_tiles[2] * VSA_TILE_SIZE[2]
    target_shape = (
        x.shape[0],
        t_padded_size * h_padded_size * w_padded_size,
```

```
            x.shape[-2],
            x.shape[-1],
        )

        # 从 metadata 中获取缓存 buffer
        buf = attn_metadata.tile_buf
        # 仅在形状 / 类型 / 设备不匹配时重新分配
        if (
            buf is None
            or buf.shape != target_shape
            or buf.dtype != x.dtype
            or buf.device != x.device
        ):
            buf = torch.zeros(target_shape, device=x.device, dtype=x.dtype)
            attn_metadata.tile_buf = buf

        # 填充非 pad 区域
        buf[:, attn_metadata.non_pad_index] = x[:, attn_metadata.tile_partition_indices]
        return buf

    # untile 方法改为使用预计算的组合索引，减少一次 fancy index
    def untile(
        self,
        x: torch.Tensor,
        untile_combined_index: torch.LongTensor,
    ) -> torch.Tensor:
        # 单次索引，替代之前的两次索引
        return x[:, untile_combined_index]
```

python/sglang/multimodal_gen/test/unit/test_video_sparse_attention.py

新增 VSA tile buffer 重用与 untile 正确性单元测试，验证缓存复用和组合索引正确性。

```
def test_video_sparse_attention_tile_buffer_reuse_and_untile():
    # 构建 metadata，使用 cpu 以便测试
    metadata = VideoSparseAttentionMetadataBuilder().build(
        current_timestep=0,
        raw_latent_shape=(5, 7, 9),
        patch_size=(1, 1, 1),
        VSA_sparsity=0.5,
        device=torch.device("cpu"),
    )

    # 创建 impl 实例，跳过 __init__ 以避免 sp_group 依赖
    impl = object.__new__(VideoSparseAttentionImpl)
    total_seq_length = metadata.total_seq_length
    x = torch.arange(2 * total_seq_length * 3 * 4, dtype=torch.float32).reshape(
        2, total_seq_length, 3, 4
    )

    # 第一次 tiling，验证 tile_buf 被设置且与返回相同引用
```

```
tiled = impl.preprocess_qkv(x, metadata)
assert metadata.tile_buf is tiled
# 验证 untile_combined_index 等于组合索引
assert torch.equal(
    metadata.untile_combined_index,
    metadata.non_pad_index[metadata.reverse_tile_partition_indices],
)
# 验证 roundtrip 正确性
assert torch.equal(impl.postprocess_output(tiled, metadata), x)

# 第二次 tiling（数据不同，但 metadata 相同），验证 buffer 被复用（data_ptr 不变）
next_x = x + 1
next_tiled = impl.preprocess_qkv(next_x, metadata)
assert next_tiled.data_ptr() == tiled.data_ptr()
assert torch.equal(impl.postprocess_output(next_tiled, metadata), next_x)

# 验证零填充区域仍然为零
pad_mask = torch.ones(next_tiled.shape[1], dtype=torch.bool)
pad_mask[metadata.non_pad_index.cpu()] = False
assert torch.all(next_tiled[:, pad_mask] == 0)
```

评论区精华

Reviewer mickqian 在 wanvideo.py 的 diff 中评论要求删除新增的 attention_type 和 sla_topk 参数及相关 del 语句（"nit: remove this"）。但最终合并时参数保留，可能为兼容 FastVideo 路径所需，该问题未实际解决。

- 移除新增的参数 attention_type 和 sla_topk (style): 参数未移除，最终合并时保留。可能为兼容 FastVideo 路径所需，但 reviewer 意见未采纳。

风险与影响

- 风险：
  1. 正确性风险：预计算的 untile_combined_index 依赖 non_pad_index 与 reverse_tile_partition_indices 的构建顺序，若后续引入条件改变构建逻辑则索引可能错位。tile buffer 复用需确保 shape、dtype、device 严格匹配，否则静默重新分配。
  2. 兼容性风险：wanvideo.py 新增的参数虽带默认值，但外部若通过关键字参数调用 super().__init__ 可能因参数名冲突受影响。
  3. 测试覆盖有限：仅一个单元测试覆盖 tile buffer 复用和 untile 正确性，未覆盖多 timestep 或不同 shape 下的复用场景。 - 影响：对使用 VSA 注意力后端的 diffusion 模型（如 Wan）有温和性能提升（显存分配次数减少、fancy index 启动减少），实测 5s 视频生成时间降低约 0.3s，峰值内存约减少 0.4 GiB。对非 VSA 模型无影响。新增测试确保基本正确性。参数添加使 Wan 模型可正常使用 VSA 配置。 - 风险标记：热点路径变更，新增参数兼容性风险，测试覆盖有限

关联脉络

- PR #1272 attention hot-path cleanup + denoising loop hoists: 本 PR 的源头，从该 PR cherry-pick 了 VSA 和 SLA 的热点清理。