PR #26973 完整报告

sgl-project/sglang

[diffusion] reduce Cosmos3 denoise overhead

合并时间：2026-06-02 14:23

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26973

执行摘要

- 一句话：Cosmos3 去噪性能优化，降低 7% 峰值内存
- 推荐动作：建议开发者关注注意力层 forward_with_replicated_kv_prefix 的设计模式，它为序列并行中处理复制前缀提供了一种低内存的拆分方案。此外，view 替代 split+contiguous 是常见的计算图优化技巧，可推广到其他类似场景。

功能与动机

当前 Cosmos3 去噪过程中存在不必要的大张量物化（如全量 K/V cat）、多次数据拷贝和每步标量同步开销，导致峰值内存高和速度受限。本 PR 针对这些热点进行优化，根据 PR body 数据，在 4xH200 上响应时间从 62.14s 降至约 61.20s，峰值内存下降约 3.8GB。

实现拆解

1. 分离 K/V 前缀与后缀，避免 Ulysses 前全量物化：在 python/sglang/multimodal_gen/runtime/layers/attention/layer.py 中新增公共方法 forward_with_replicated_kv_prefix，接收分开的 k_prefix/v_prefix 和 k_suffix/v_suffix。内部根据 SP 配置选择是否走 Ulysses all-to-all，并最终调用 _forward_with_replicated_kv_prefix_split，该方法将 all-to-all 分别作用于 Q 与 K/V 后缀，再在局部切片后连接前缀，避免在全量 K/V 上执行 all-to-all。原有 _forward_with_replicated_kv_prefix（接受已 cat 的 K/V）改为委托调用新 split 方法，保持向后兼容。
2. 使用 view + slicing 替换 split + contiguous：在 cosmos3video.py 的 Cosmos3SelfAttention.forward 和 Cosmos3CrossAttention.forward 中，将 qkv.split(...) 后逐个 .contiguous().view(...) 替换为直接 .view(...) 后切片，减少一次显式拷贝（前提是 qkv 来自线性层输出，内存连续）。
3. 缓存局部分片 RoPE：在 cosmos3video.py 的语言模型前向后，解包 freqs_gen 为 cos_gen, sin_gen，若启用序列分片则做 padding 后缓存，避免每步切片计算。
4. 使用循环步索引代替 timestep 标量同步：在 cosmos3.py 的 denoise 循环中，将当前的步索引 i 作为 current_timestep 参数传入 _run_transformer 及其所有调用链（_predict_noise_cfg_parallel、_predict_noise_cfg_batched），替代每步执行 int(timestep.flatten()[0].item()) 的同步开销。_run_transformer 增加可选参数，优先使用传入值。

5. 其他配套调整：Cosmos3CrossAttention.forward 直接调用新 attention 接口；cached_freqs_gen 存储逻辑重构以支持分片 RoPE 缓存。

关键文件：

- python/sglang/multimodal_gen/runtime/layers/attention/layer.py（模块 注意力层；类别 source；类型 core-logic；符号 forward_with_replicated_kv_prefix, _forward_with_replicated_kv_prefix_split）：核心注意力层变更：新增公共接口 forward_with_replicated_kv_prefix，分离 K/V 前缀与后缀，避免全量物化，并重构原有内部方法。
- python/sglang/multimodal_gen/runtime/models/dits/cosmos3video.py（模块 模型定义；类别 source；类型 data-contract；符号 Cosmos3SelfAttention.forward, Cosmos3CrossAttention.forward, Cosmos3GENVideoTransformer.forward）：Cosmos3 注意力层实现变更：自注意力使用 view 替代 split+contiguous；交叉注意力通过新接口分离 K/V；缓存局部 RoPE 分片。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py（模块 管线逻辑；类别 source；类型 data-contract；符号 _run_transformer, _predict_noise_cfg_parallel, _predict_noise_cfg_batched）：Pipeline 层优化：引入 current_timestep 参数避免每步标量同步，并传递到所有 CFG 路径。

关键符号：forward_with_replicated_kv_prefix, _forward_with_replicated_kv_prefix_split, Cosmos3SelfAttention.forward, Cosmos3CrossAttention.forward, Cosmos3GENVideoTransformer.forward, _run_transformer, _predict_noise_cfg_parallel, _predict_noise_cfg_batched

关键源码片段

python/sglang/multimodal_gen/runtime/layers/attention/layer.py

核心注意力层变更：新增公共接口 forward_with_replicated_kv_prefix，分离 K/V 前缀与后缀，避免全量物化，并重构原有内部方法。

```
def forward_with_replicated_kv_prefix(
    self,
    q: torch.Tensor,
    k_prefix: torch.Tensor,
    v_prefix: torch.Tensor,
    k_suffix: torch.Tensor,
    v_suffix: torch.Tensor,
) -> torch.Tensor:
    '''attention with replicated K/V prefix supplied separately

    该接口将 K/V 分为前缀（跨 rank 复制）和后缀（按序列分片），
    避免在 all-to-all 前物化完整 K/V 张量。
    '''
    forward_context: ForwardContext = get_forward_context()
    ctx_attn_metadata = forward_context.attn_metadata

    # 非 SP 模式：直接 concat 后调用底层实现
```

```
    if self.skip_sequence_parallel or get_sequence_parallel_world_size() == 1:
        k = torch.cat([k_prefix, k_suffix], dim=1)
        v = torch.cat([v_prefix, v_suffix], dim=1)
        return self.attn_impl.forward(q, k, v, ctx_attn_metadata)

    # 只有 TP/DP，无 Ulysses：同样 concat 后走默认 forward
    if get_ulysses_parallel_world_size() == 1:
        k = torch.cat([k_prefix, k_suffix], dim=1)
        v = torch.cat([v_prefix, v_suffix], dim=1)
        return self(q, k, v)

    # Ulysses 并行：使用 split 形式避免大 K/V 物化
    return self._forward_with_replicated_kv_prefix_split(
        q, k_prefix, v_prefix, k_suffix, v_suffix, ctx_attn_metadata
    )


def _forward_with_replicated_kv_prefix_split(
    self,
    q: torch.Tensor,
    k_rep: torch.Tensor,
    v_rep: torch.Tensor,
    k_shard: torch.Tensor,
    v_shard: torch.Tensor,
    ctx_attn_metadata,
) -> torch.Tensor:
    '''split form 避免在 Ulysses all-to-all 前物化完整 K/V'''
    sp_rank = get_sp_parallel_rank()

    # 1. all-to-all 将 Q 和 K/V 后缀从序列分片转到头分片
    q = _usp_input_all_to_all(q, head_dim=2)
    k_shard = _usp_input_all_to_all(k_shard, head_dim=2)
    v_shard = _usp_input_all_to_all(v_shard, head_dim=2)

    # 2. 将复制的前缀切片到与本 rank 的头分片一致
    h_kv_local = k_shard.shape[2]
    h_start = sp_rank * h_kv_local
    h_end = h_start + h_kv_local
    k_rep = k_rep[:, :, h_start:h_end, :].contiguous()
    v_rep = v_rep[:, :, h_start:h_end, :].contiguous()

    # 3. 在序列维度 cat 前缀和后缀，执行本地 attention
    k = torch.cat([k_rep, k_shard], dim=1)
    v = torch.cat([v_rep, v_shard], dim=1)

    out = self.attn_impl.forward(q, k, v, ctx_attn_metadata)
    # 4. all-to-all 将输出从头分片转回序列分片
    return _usp_output_all_to_all(out, head_dim=2)
```

python/sglang/multimodal_gen/runtime/models/dits/cosmos3video.py

Cosmos3 注意力层实现变更：自注意力使用 view 替代 split+contiguous；交叉注意力通过新接口分离 K/V；缓存局部 RoPE 分片。

```
# 在 Cosmos3CrossAttention.forward 中，从 QKV 投影直接 view 切片
qkv, _ = self.to_qkv(hidden_states)
# 使用 view + slicing 替代 split + contiguous，减少拷贝
qkv = qkv.view(
    batch_size, seq_len_gen,
    self.num_attention_heads + 2 * self.num_key_value_heads,
    self.head_dim,
)
q = qkv[:, :, : self.num_attention_heads, :]
k = qkv[:, :, self.num_attention_heads : self.num_attention_heads + self.num_key_value_heads, :
]
v = qkv[:, :, self.num_attention_heads + self.num_key_value_heads :, :]

# 应用 QK norm 和 RoPE
q = F.rms_norm(q, (self.head_dim,), self.norm_q.weight, self.norm_q.variance_epsilon)
k = F.rms_norm(k, (self.head_dim,), self.norm_k.weight, self.norm_k.variance_epsilon)
q, k = qwen3_apply_rotary_pos_emb(q, k, freqs_cos, freqs_sin)

# 直接调用新 attention 接口，分别传入前缀 (k_und, v_und) 和后缀 (k, v)
out = self.attn.forward_with_replicated_kv_prefix(q, k_und, v_und, k, v)
out = out.reshape(batch_size, seq_len_gen, -1)
out, _ = self.to_out(out)
return out
```

评论区精华

本 PR 无 Review 评论或讨论，仅有机器人自动评论和 /tag-and-rerun-ci 操作。

- 暂无高价值评论线程

风险与影响

- 风险：
  - view 内存连续性假设：替换 split+contiguous 为 view 要求 qkv 在内存上连续，通常满足；若模型量化或特殊层引入非连续内存，可能触发 view 失败。已通过位一致输出验证，风险可控。
  - current_timestep 等价性：循环索引 i 必须严格对齐实际 timestep 顺序，若未来改动 denoise 循环（如异步或乱序），该假设可能不成立。当前验证输出一致。
  - 新接口覆盖度：forward_with_replicated_kv_prefix 目前仅被 Cosmos3CrossAttention 使用，若后续其他模型共用需确认接口契约兼容。
  - 影响：影响范围：仅限于 diffusion 子系统中的 Cosmos3 模型推理。去噪阶段响应时间降低约 1.5%（~0.9s），峰值内存降低约 7%（~3.8GB）。对其他模型（如 WanVideo）无影响。注意力层新增公共方法对其他用户透明，旧接口仍可用。

- 风险标记：view 内存连续性依赖，新接口仅被单模型使用

关联脉络

- 暂无明显关联 PR