PR #28319 完整报告

sgl-project/sglang

[diffusion] Shard HunyuanVideo text tokens under SP

合并时间：2026-06-20 18:21

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28319

执行摘要

- 一句话：在序列并行下对 HunyuanVideo 文本 token 进行分片，降低通信开销
- 推荐动作：该 PR 是针对特定分布式配置的有效性能优化，设计清晰（条件判定 + 通道分离），值得关注其变长通信原语的复用潜力。建议阅读 layer.py 中的 seq_lens 分支实现。

功能与动机

原有实现中，即使启用序列并行，文本 token 依然在每个 GPU 上完整复制并参与 attention 的全 gather，造成通信和计算冗余。PR body 指出目标是在 ring_degree=1 时对文本 token 进行分片，以消除复制冗余，降低 all-to-all 通信量。

实现拆解

1. 分片条件判定：在 hunyuanvideo.py 的 HunyuanVideo.forward 中，当 sp_size > 1、ring_degree == 1、文本序列长度不小于 sp_size 且不在梯度计算中时，将 txt_is_sharded 设置为 True，并计算每个 rank 的文本分片长度 text_seq_lens。
2. 分片数据准备：根据 sp rank 将完整文本 tensor 切片为本地 shard，并将分片后的图像 + 文本在序列维度拼接。
3. 变长 all-to-all 通信：在 layer.py 的 UlyssesAttention.forward 中，当传入 seq_lens 参数时，使用新增的 _usp_input_all_to_all_varlen 和 _usp_output_all_to_all_varlen 替代原先的等长 all-to-all，支持不均匀序列长度的头维度分散 / 收集。
4. 注意力结果恢复：在 hunyuanvideo.py 的双流 / 单流块中，分片模式下 self.attn 返回拼接后的输出，再按图片和文本长度切分回各自 tensor。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/hunyuanvideo.py（模块 扩散模型；类别 source；类型 core-logic；符号 DualBlock.forward, SingleBlock.forward, HunyuanVideo.forward）：核心模型文件，新增分片条件判定、分片长度计算，以及双流 / 单流块的分片推理路径。
- python/sglang/multimodal_gen/runtime/layers/attention/layer.py（模块 注意力层；类别 source；类型 core-logic；符号 UlyssesAttention.forward）：注意力层基类，新增变长 all-to-all 通信原语，支持不均匀序列长度的 Ulysses 注意力。

关键符号：DualBlock.forward, SingleBlock.forward, HunyuanVideo.forward, UlyssesAttention.forward

关键源码片段

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

核心模型文件，新增分片条件判定、分片长度计算，以及双流 / 单流块的分片推理路径。

```
def forward(
    self,
    img: torch.Tensor,
    txt: torch.Tensor,
    vec: torch.Tensor,
    freqs_cis: tuple,
    txt_is_sharded: bool = False, # 新增：标记文本是否已分片
    seq_lens: list[int] | None = None, # 新增：各 rank 的文本序列长度
) -> tuple[torch.Tensor, torch.Tensor]:
    # ... 省略调制、QKV 计算等前序逻辑 ...
    if txt_is_sharded:
        # 分片模式下将图像和文本拼接后一次性送入 attention，由 UlyssesAttention 内部处理变长 all-to-all
        attn, _ = self.attn(
            torch.cat((img_q, txt_q), dim=1),
            torch.cat((img_k, txt_k), dim=1),
            torch.cat((img_v, txt_v), dim=1),
            seq_lens=seq_lens,
        )
        img_attn, txt_attn = attn.split([image_seq_len, text_seq_len], dim=1)
    else:
        # 非分片模式：保留原有双输出路径（完整复制文本，分别计算并返回）
        img_attn, txt_attn = self.attn(img_q, img_k, img_v, txt_q, txt_k, txt_v)
    # ... 后续投影、残差连接 ...
```

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

注意力层基类，新增变长 all-to-all 通信原语，支持不均匀序列长度的 Ulysses 注意力。

```
def forward(
    self,
    q, k, v,
    replicated_q=None, replicated_k=None, replicated_v=None,
    seq_lens: list[int] | None = None, # 新增：每个 rank 的原始序列长度（用于变长 all-to-all）
) -> tuple[torch.Tensor, torch.Tensor | None]:
    # ... 形状检查、上下文获取 ...
    if seq_lens is not None:
        # 变长路径：禁止 replicated QKV（文本分片场景不使用）
        assert replicated_q is None and replicated_k is None and replicated_v is None
    # 堆叠 QKV
    qkv = torch.cat([q, k, v], dim=0)
    if seq_lens is None:
```

```
        # 原始等长 all-to-all：按 head 维度 scatter，序列维度 gather
        qkv = sequence_model_parallel_all_to_all_4D(qkv, scatter_dim=2, gather_dim=1)
    else:
        # 变长 all-to-all：根据 seq_lens 计算偏移，进行 head 维均匀 scatter 和序列维动态 gather
        qkv = _usp_input_all_to_all_varlen(qkv, seq_lens, head_dim=2)
    # ... preprocess, attention compute ...
    if seq_lens is None:
        output = sequence_model_parallel_all_to_all_4D(output, scatter_dim=1, gather_dim=2)
    else:
        output = _usp_output_all_to_all_varlen(output, seq_lens, head_dim=2)
    return output, replicated_output
```

评论区精华

无实质 review 讨论，仅有自动 ci 触发评论和 gemini 配额警告。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 正确性风险：条件 torch.is_grad_enabled() 阻止了训练场景，但推理时若文本序列过短（< sp_size）则不会分片，逻辑安全。
  2. 性能风险：分片后每个 rank 处理的文本长度不均衡（当文本无法被 sp_size 整除时新增了余数分发），可能引入轻微负载不均。但 PR 基准测试显示总体正向收益。
  3. 兼容性：仅影响 UlyssesAttention 路径，且通过 seq_lens 参数选择旧 / 新路径，无默认行为破坏。
  4. 测试覆盖：该 PR 未附带新测试，回归风险存在。 - 影响：对用户：在 4+ GPU 且启用 SP (--ulysses-degree >=2 --ring-degree 1) 时自动获得 3-5% 推理加速；无需改代码。对开发者：新增的变长 all-to-all 通信原语（_usp_input_all_to_all_varlen、_usp_output_all_to_all_varlen）可复用于其他扩散模型的分布式分片。 - 风险标记：变长通信正确性依赖 seq_lens 计算，仅 ulysses 路径生效，ring 路径无影响，未附带测试

关联脉络

- 暂无明显关联 PR