PR #30107 完整报告

sgl-project/sglang

[diffusion] perf: add unified SP shard helpers and zero-copy tail-pad attention

合并时间：2026-07-04 23:57

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30107

执行摘要

- 一句话：统一 diffusion SP 切分逻辑并引入零拷贝尾部填充注意力
- 推荐动作：值得精读的设计决策：通过布局不变性将 padding 开销消除在注意力之前，无需自定义 kernel，展示了如何利用现有 kernel（FlashAttention varlen）实现零成本 padding。适合有序列并行需求的团队参考。建议关注：SpShard 数据类、tail_attn_meta 的构造、shard_like 的 pad_mode 分支。

功能与动机

原来存在四个独立 pad 实现，语义各异且其中两个存在 bug：mova 的视频 / 音频 tower 使用无 mask 零填充，pad token 作为 K/V 进入 softmax 污染每个真实 token（MAE 约 1.2e-2 当 1 pad / 16 tokens）；base.shard_latents_for_sp 在不可分维上同样无 mask。需要统一到正确不变性并消除重复代码。

实现拆解

1. 创建统一 SP 切分模块（sp_shard_utils.py）：定义 SpShard 数据类（含 local_pad、local_real_len 属性）和核心函数 build_shard_plan、shard_like、shard_seq、gather_seq、tail_attn_meta 等。布局不变性：填充始终在最后一个 rank 的本地块尾部，全局序列尾部为一个连续 pad 块。
2. 迁移模型 shard 逻辑：将 qwen_image.py、flux.py、flux_2.py 中的 _shard_text_for_sp 和 _pad_shard_text_for_sp_varlen 实现，替换为调用新模块中的 build_shard_plan、shard_like、join_seqs/split_seqs 等，消除三份字节相同的 gap 实现。
3. 修复 mova 和 base 的 mask bug：mova 的 _shard_sequence_for_sp 和 _gather_sequence_from_sp 使用新模块，自动获得零填充尾部布局；ernie_image 跳过 SP 直到流真正被 shard。
4. USPAttention 零拷贝尾部路径：在 layer.py 中，当检测到尾部分布（通过 tail_attn_meta），直接将 padded 布局输入 varlen FA，每个 batch row 拆分为 [valid | pad] 段，仅做连续 reshape，无 repacking cat 或索引 gather。
5. 默认 shard 策略：通过 bench 确认 shard 在所有文本长度均优于 replicate，故默认 always-shard；提供 SGLANG_SP_TEXT_SHARD_MIN 环境变量作为逃生口。
6. 测试配套：新增 test_sp_shard.py 含 17 个单元测试（shard 数学、尾部分布、策略门控、gather/trim）。更新 h100.json 中 mova 一致性阈值（因 bugfix 改变行为）。

关键文件：

- python/sglang/multimodal_gen/runtime/distributed/sp_shard_utils.py（模块 分布式；类别 source；类型 core-logic；符号 SpShard, local_pad, local_real_len, build_shard_plan）：核心新增模块，定义统一 SP 切分抽象，包含布局不变性、shard 数学、尾部 attention 元数据生成。
- python/sglang/multimodal_gen/runtime/models/dits/qwen_image.py（模块 模型层；类别 source；类型 data-contract；符号 _shard_text_for_sp, _pad_shard_text_for_sp_varlen）：大规模重构，删除 124 行旧 shard 逻辑，替换为调用新模块函数。
- python/sglang/multimodal_gen/runtime/models/dits/flux_2.py（模块 模型层；类别 source；类型 data-contract；符号 _shard_text_for_sp）：同 qwen_image，重构 shard 逻辑，删除 144 行旧代码。
- python/sglang/multimodal_gen/runtime/models/dits/flux.py（模块 模型层；类别 source；类型 data-contract；符号 _shard_text_for_sp）：同 flux_2，重构 shard 逻辑，删除 127 行旧代码。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/mova.py（模块 流水线；类别 source；类型 data-contract）：修复 mova 的 mask bug，迁移到新 shard 模块。
- python/sglang/multimodal_gen/test/unit/test_sp_shard.py（模块 测试；类别 test；类型 test-coverage；符号 _fake_sp, test_plan_shard_divisible, test_plan_shard_padded_last_rank, test_shard_like_zero_pads_tail）：新增单元测试，覆盖 shard 数学、尾部元数据、策略门控、gather/trim 等 17 个场景。
- python/sglang/multimodal_gen/runtime/layers/attention/layer.py（模块 注意力层；类别 source；类型 core-logic）：实现零拷贝尾部注意力路径，当检测到尾部分布时直接使用 varlen FA 跳过 pad。

关键符号：SpShard, local_pad, local_real_len, build_shard_plan, shard_like, shard_seq, gather_seq, shard_seq_prefix, tail_attn_meta, should_shard_text, join_seqs, split_seqs, _shard_text_for_sp, _pad_shard_text_for_sp_varlen, _shard_sequence_for_sp, _gather_sequence_from_sp

关键源码片段

python/sglang/multimodal_gen/runtime/distributed/sp_shard_utils.py

核心新增模块，定义统一 SP 切分抽象，包含布局不变性、shard 数学、尾部 attention 元数据生成。

```
# SPDX-License-Identifier: Apache-2.0
"""Unified SP shard / pad / gather helpers.

Layout invariant: padding always sits at the end of the LAST rank's local
chunk, so the gathered sequence has one contiguous pad block at its global
tail. `tail_attn_meta` then lets attention skip it for free (the pad becomes
its own varlen segment — no repacking, no mask compute).
```

```
"""

import torch
import torch.nn.functional as F

from sglang.multimodal_gen.runtime.distributed.parallel_state import (
    get_sp_parallel_rank, get_sp_world_size,
)



@dataclass(frozen=True)
class SpShard:
    """Facts of one tail-padded even shard, shared by tensors of that stream."""
    orig_len: int # real tokens (global)
    local_len: int # per-rank chunk length (equal on every rank)
    num_pad: int # pad tokens, all at the last rank's local tail
    sp_size: int
    sp_rank: int

    @property
    def local_pad(self) -> int:
        """Pad rows inside THIS rank's chunk (tail of the last rank)."""
        return self.num_pad if self.sp_rank == self.sp_size - 1 else 0

    @property
    def local_real_len(self) -> int:
        """Real tokens in this rank (excluding pad)."""
        return self.local_len - self.local_pad



def build_shard_plan(seq_len: int) -> SpShard:
    """Pure shard math; no tensor operations."""
    sp_size = get_sp_world_size()
    if sp_size <= 1:
        return SpShard(seq_len, seq_len, 0, 1, 0)
    local_len = (seq_len + sp_size - 1) // sp_size
    return SpShard(
        orig_len=seq_len,
        local_len=local_len,
        num_pad=local_len * sp_size - seq_len,
        sp_size=sp_size,
        sp_rank=get_sp_parallel_rank(),
    )



def tail_attn_meta(
    shard: SpShard, seqlen_multiplier: int, device: torch.device
) -> Optional[dict]:
    """Build varlen metadata for USPAttention to skip the tail pad block.
```

```
    Returns None if no pad exists.
    """
    if shard.num_pad == 0:
        return None
    valid = shard.local_real_len * seqlen_multiplier
    # ensure max_seqlen >= num_pad to satisfy FlashAttention contract
    return {
        "tail": True,
        "valid_len": valid,
        "max_seqlen_tail": max(valid, shard.num_pad),
    }


def shard_like(
    x: torch.Tensor, shard: SpShard, dim: int = 1, pad_mode: str = "zeros"
) -> torch.Tensor:
    """Apply a planned shard to one tensor (e.g., RoPE cache must use same
    plan as hidden states to keep chunks aligned)."""
    if shard.sp_size <= 1:
        return x
    if shard.num_pad > 0:
        if pad_mode == "repeat_last":
            pad = x.narrow(dim, x.shape[dim] - 1, 1).expand(
                *[shard.num_pad if i == dim else -1 for i in range(x.dim())]
            )
            x = torch.cat([x, pad], dim=dim)
        else:
            # F.pad dimension order: (left, right) pairs from last dim
            pads = [0, 0] * (x.dim() - 1 - dim) + [0, shard.num_pad]
            x = F.pad(x, pads)
    return x.narrow(dim, shard.sp_rank * shard.local_len, shard.local_len)
```

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

大规模重构，删除 124 行旧 shard 逻辑，替换为调用新模块函数。

```
# 从 sp_shard_utils 导入新函数
from sglang.multimodal_gen.runtime.distributed.sp_shard_utils import (
    build_shard_plan,
    join_seqs,
    shard_like,
    should_shard_text,
    split_seqs,
    tail_attn_meta,
)

# 旧 _shard_text_for_sp / _pad_shard_text_for_sp_varlen 被完全删除
# 新的调用方式示例（在 forward 中使用）：
def forward(self, ...):
    text_shard = build_shard_plan(encoder_hidden_states.shape[1])
```

```
if text_shard.num_pad > 0:
    attn_meta = tail_attn_meta(text_shard, 1, device)
    encoder_hidden_states = shard_like(encoder_hidden_states, text_shard)
    # 注意的 joint 序列使用 split_seqs/join_seqs 将 pad 移到尾部
    joint_q, joint_k, joint_v = join_seqs(
        [txt_q, txt_k, txt_v], [img_q, img_k, img_v],
        local_pad=text_shard.local_pad
    )
```

评论区精华

核心讨论来自 gemini-code-assist[bot] 的高优先级评论：tail_attn_meta 中的 max_seqlen_tail 在 num_pad > valid 时会传入小于填充长度的值，违反 FlashAttention 的契约，可能导致 CUDA 未定义行为或静默崩溃。作者接受并以 max(valid, shard.num_pad) 修复（commit e6e2525），并补充了退化用例（orig_len=1, sp=4）的单元测试。

- max_seqlen_tail 安全隐患：当 num_pad > valid 时传入小于填充长度的值可能违反 FlashAttention 契约 (correctness): 作者接受并改用 max(valid, shard.num_pad)，在 commit e6e2525 中修复，并添加退化用例测试。

风险与影响

- 风险：
  1. 回归风险：新模块替换了多个模型的 shard 逻辑，尽管有单元测试覆盖 17 个核心场景，但集成测试（如端到端 DiT forward）仅覆盖 QwenImage，其他模型需额外验证。
  2. FlashAttention varlen 依赖：零拷贝尾部路径依赖 FA varlen 的正确性，如果 FA 后端有差异可能导致溢出或错误。
  3. 默认 shard 策略：always-shard 在 ring > 1 时 fallback 到 replicate，但无 mask 的 replicate 路径（mova 旧行为）已修复；但仍需确认 ring 场景无退化。
  4. 行为变更：mova 的 pad mask 从无到有改变了数值结果（一致性测试阈值已更新），依赖旧行为的用户需注意。
  5. RoPE cache 重排：shard_seq_prefix 和 join_seqs 对 RoPE cache 的重新排序可能影响非注意力模块的数值。 - 影响：影响范围：diffusion 模块下所有使用序列并行的模型（flux、flux_2、qwen_image、mova、ernie_image）。影响程度：注意力部分延迟降低 4–8%，修复两个可能导致静默数值错误的关键 bug，代码量减少约 200 行。用户只需升级版本即可获得性能和正确性提升，无需修改调用代码。系统影响：零拷贝路径不改变 API，仅内部布局变化。 - 风险标记：新统一 SP 模块，注意力零拷贝路径，mask 行为变化，默认 shard 策略变更

关联脉络

- 暂无明显关联 PR