PR #32667 完整报告

sgl-project/sglang

[Diffusion] Add K/V-gather sequence parallel attention

合并时间：2026-08-07 09:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32667

执行摘要

- 一句话：为 Diffusion 新增 K/V-gather 序列并行注意力并默认在 SP2 启用
- 推荐动作：值得精读，尤其是三个设计决策：mode 到 degree 的接口演进、auto 默认值与 fail-closed 的边界划分、以及以实测数据矩阵驱动默认策略的方法。对于需要部署 Diffusion SP 的团队，建议先仔细阅读文档中的拓扑权衡表，并在目标硬件上复测 SP2 与高 degree 场景。

功能与动机

PR body 明确指出："No single exchange wins across the measured models and topologies"，现有 Ulysses all-to-all 在部分拓扑和 torch.compile 场景下并非最优。K/V-gather 的每卡通信量约为 Ulysses 的 P/2 倍，但它能减半 collective 数量并移除输出交换，在 SP2 及部分高 degree 场景下实测更优。因此需要一个可独立配置的 SP 维度，并基于实测数据做默认值决策。

实现拆解

实现拆解如下：

1. 新增 SP 维度与参数体系：在 python/sglang/multimodal_gen/runtime/server_args/server_args.py 中新增 kv_gather_degree 与 sp_split_auto 字段，CLI 暴露 --kv-gather-degree。SP 关系式变为 sp_degree = ulysses_degree x ring_degree x kv_gather_degree；_adjust_parallelism 在无显式 SP 参数且 sp_degree=2 时自动赋 kv_gather_degree=2，否则保持 Ulysses 默认；_validate_parallelism 校验 kv_gather_degree 必须等于 sp_degree，且暂不支持与 Ulysses/Ring 组合，显式组合直接报错。
2. 注意力层核心实现：在 python/sglang/multimodal_gen/runtime/layers/attention/layer.py 中新增 _resolve_sp_attention_mode 按层解析交换模式；_forward_with_kv_gather 对 K/V 执行序列维 all-gather 后本地计算注意力，支持 replicated token 拼接与输出切分；_kv_gather_unsupported_reason 检测 pre-a2a、多 replicated 模式、mask 形态等不支持的调用，显式模式 fail-closed、自动模式逐调用回退 Ulysses。同时扩展 UlyssesAttention 与 USPAttention 的 forward 分支，并对 UlyssesAttention_VSA 直接拒绝视频稀疏注意力。

3. 模型兼容修正：python/sglang/multimodal_gen/runtime/models/dits/qwen_image.py 中 segmented pre-all-to-all 仅保留在 Ulysses 模式；K/V-gather 下 Qwen 走 join_seqs 路径，避免 Ulysses 布局与行切分布局错配导致的 matmul 形状错误。python/sglang/multimodal_gen/runtime/distributed/sp_shard_utils.py 中 plan_text_strategy 增加 padding 必须落在最后一个 shard 的约束，否则改为 replicate。
4. 测试与文档：新增 test_usp_attention_kv_gather.py，覆盖本地 Q 对 gathered KV 的数值正确性、replicated prefix/suffix、padding mask、varlen/ 视频稀疏拒绝、模式解析与 fail-closed 行为；test_server_args.py 新增 TestKVGatherDegree 覆盖默认分配、显式不覆盖、组合报错等；test_sp_shard.py 覆盖 padding 跨 shard 的 replicate 策略。文档 ring_sp_performance.mdx 大幅扩充，说明三种交换方式的通信量、内存和拓扑权衡，并给出 TP+SP、FSDP+SP 的 benchmark 数据。

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 _resolve_sp_attention_mode, _kv_gather_unsupported_reason, _forward_with_kv_gather, _gather_sharded_sequence）：核心实现文件，新增 K/V-gather 交换路径、模式解析与 fail-closed 逻辑，改动量最大。
- python/sglang/multimodal_gen/runtime/server_args/server_args.py（模块 服务参数；类别 source；类型 core-logic）：定义新 SP 维度参数、默认分配逻辑与组合校验，是行为开关所在。
- python/sglang/multimodal_gen/test/unit/test_usp_attention_kv_gather.py（模块 单元测试；类别 test；类型 test-coverage；符号 _SdpaAttention, _make_attention, _reference_attention, TestUSPAttentionKVGather）：新增的完整测试套件，覆盖数值正确性、mask、replicated token、拒绝路径与模式解析。
- python/sglang/multimodal_gen/runtime/distributed/sp_shard_utils.py（模块 分片工具；类别 source；类型 core-logic）：修复 SP 文本 padding 跨多个 shard 时的布局问题，避免 K/V-gather 路径下 padding 错位。
- python/sglang/multimodal_gen/runtime/models/dits/qwen_image.py（模块 模型适配；类别 source；类型 data-contract）：适配 K/V-gather 布局，避免 segmented pre-a2a 在 kv_gather 模式下产生形状不匹配。

关键符号：_resolve_sp_attention_mode, _forward_with_kv_gather, _kv_gather_unsupported_reason, _gather_sharded_sequence, _adjust_parallelism

关键源码片段

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

核心实现文件，新增 K/V-gather 交换路径、模式解析与 fail-closed 逻辑，改动量最大。

```
def _resolve_sp_attention_mode(*, causal: bool, sparse_backend: bool) -> tuple[str, bool]:
    """解析某一层的 SP 交换方式，返回 (mode, is_auto)。

    kv_gather_degree > 1 时选择 gather 交换；若 degree 为自动分配，
    无法服务的层回退到 Ulysses；显式指定时则 fail-closed。
```

```
    """
    from sglang.multimodal_gen.runtime.server_args import get_global_server_args

    args = get_global_server_args()
    if args.kv_gather_degree <= 1:
        return "ulysses", False
    if causal or sparse_backend:
        if args.sp_split_auto:
            return "ulysses", True # 自动模式下回退，保留 is_auto 标记
        if causal:
            raise ValueError("K/V-gather SP does not support causal attention.")
        raise NotImplementedError("K/V-gather SP does not support sparse attention backends.")
    return "kv_gather", args.sp_split_auto


def _forward_with_kv_gather(
    self,
    q: torch.Tensor,
    k: torch.Tensor,
    v: torch.Tensor,
    ctx_attn_metadata,
    replicated_q: torch.Tensor | None,
    replicated_k: torch.Tensor | None,
    replicated_v: torch.Tensor | None,
    seq_lens: list[int] | None,
) -> tuple[torch.Tensor, torch.Tensor | None]:
    # 只支持非 varlen 路径，与 Ulysses 的 varlen a2a 互斥
    if seq_lens is not None:
        raise NotImplementedError("K/V-gather SP does not support varlen UlyssesAttention.")
    # replicated Q/K/V 必须成组出现
    if any(x is not None for x in (replicated_q, replicated_k, replicated_v)):
        if any(x is None for x in (replicated_q, replicated_k, replicated_v)):
            raise ValueError("Replicated Q, K, and V must be provided together.")

    # 关键交换：只 all-gather K/V，不需要 Q 的 a2a
    k = sequence_model_parallel_all_gather(k, dim=1)
    v = sequence_model_parallel_all_gather(v, dim=1)

    local_query_len = q.shape[1]
    if replicated_q is not None:
        # 本地 Q 与 replicated token 拼接后一并计算
        q = torch.cat([q, replicated_q], dim=1)
        k = torch.cat([k, replicated_k], dim=1)
        v = torch.cat([v, replicated_v], dim=1)

    output = self.attn_impl.forward(q, k, v, ctx_attn_metadata)
    if replicated_q is None:
        return output, None
    # 按 local / replicated 切分输出，保持与 Ulysses 路径一致的接口
```

```
    return output[:, :local_query_len], output[:, local_query_len:]
```

python/sglang/multimodal_gen/runtime/server_args/server_args.py

定义新 SP 维度参数、默认分配逻辑与组合校验，是行为开关所在。

```
# 在 _adjust_parallelism 中，原来只自动设置 ulysses_degree，现在按 SP degree 区分：
if (
    self.ulysses_degree is None
    and self.ring_degree is None
    and self.kv_gather_degree is None
    and self.sp_degree != 1
):
    if self.sp_degree == 2:
        # 2 路 SP 是 K/V-gather 实测稳定获益区间，默认启用并标记为自动分配
        self.kv_gather_degree = 2
        self.sp_split_auto = True
        logger.info(
            "Automatically set kv_gather_degree=sp_degree=2; set "
            "--ulysses-degree explicitly to keep the Ulysses exchange"
        )
    else:
        # 更高 degree 下 K/V-gather 通信量放大，仍默认 Ulysses
        self.ulysses_degree = self.sp_degree
        logger.info(
            "Automatically set ulysses_degree=sp_degree=%d for the "
            "sequence-parallel process-group layout",
            self.ulysses_degree,
        )

if self.kv_gather_degree is None:
    self.kv_gather_degree = 1

if self.kv_gather_degree > 1:
    # 尚未实现与 Ulysses / Ring 的组合，显式组合直接报错
    if (self.ulysses_degree or 1) != 1 or (self.ring_degree or 1) != 1:
        raise ValueError(
            "kv_gather_degree does not compose with ulysses_degree or "
            "ring_degree yet; set exactly one of them above 1"
        )
```

评论区精华

本 PR 没有外部 review 评论，但作者在 Issue 评论中记录了多次设计迭代：早期曾以 --sp-attention-mode 作为模式开关，后经设计评审改为一等 SP 维度 --kv-gather-degree。作者说明理由：gather 切分的是行（与 ring 同维度），而不是 Ulysses 切分的头维度，若用 mode 复用 --ulysses-degree 会让同一参数含义随上下文变化；改为 degree 后各维度语义固定，未实现的组合也能以清晰报错表达。另一关键决策是默认值策略完全由实测数据驱动：SP2 是唯一全模型一致获益区间，故仅在该区间自动启用，并保留 sp_split_auto 让无法服务

的层回退；显式设置则 fail-closed。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险集中在：1）layer.py 的核心 attention 路径变更，影响所有 Diffusion SP 推理，auto 模式下用户可能无感知地切换到 K/V-gather 路径，若新路径存在未覆盖的调用形态可能回退到 Ulysses，行为差异不易察觉；2）K/V 在每个 SP rank 内复制，注意力激活内存高于 Ulysses，长序列（如 LTX 241 帧）场景可能 OOM，PR 中也确认了 2x H100 下默认分配器需配合 expandable_segments 才能稳定跑完；3）qwen_image.py 强制切换 join_seqs 路径可能引入数值差异或性能回退；4）性能结论强依赖硬件与 compile 配置，8x H200 与 4x H100 的交叉点不一致，auto 规则可能在其他架构上不是最优；5）sp_shard_utils.py 的 padding 策略收紧可能使部分短文本从 shard 改为 replicate，增加通信量。
- 影响：影响范围为 SGLang Diffusion 的序列并行推理：2 卡无显式 SP 参数的用户默认切换到 K/V-gather 交换，延迟在实测模型中普遍改善约 2-3%，但内存峰值可能上升；SP degree > 2 默认行为不变。对开发者而言，新增了一个并行维度，需要理解 ulysses_degree 与 kv_gather_degree 的独占关系；后续要支持 kv_gather x ulysses 或 kv_gather x ring 组合时，需要重构进程组构造。文档层面为 ring_sp_performance.mdx 增加了大量可复用的 benchmark 参考，有助于用户按拓扑选型。
- 风险标记：核心路径变更，默认行为变化，硬件相关性能波动，未实现组合报错，K/V 复制内存开销增加

关联脉络

- PR #33850 [diffusion] retire released warmup and decoder flags: 同样修改了 multimodal_gen/runtime/server_args/server_args.py，属于同一模块的参数面持续演进。
- PR #33843 [diffusion] consolidate pipeline core hygiene: 同样修改了 server_args.py 与 pipeline 核心，与本 PR 在服务参数层有交集。