

PR #39291 完整报告

sgl-project/sglang

[Diffusion] Make the SP sequence gather pass contiguous shards

合并时间: 2026-09-13 20:30

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39291>

执行摘要

- 一句话: 修复 SP 序列收集批量时非连续分片崩溃
- 推荐动作: 值得精读。这是一个小而精准的 bugfix, 根因分析透彻, 修复方式仅用 `.contiguous()` 两处微调就消除了默认配置下的启动崩溃, 且测试设计具有示范性: 用 `mock all-gather` 在单进程内捕获 contiguity 问题, 避免了多卡测试成本。适合作为「PyTorch 张量连续性陷阱 + 集合通信」相关问题的参考案例。

功能与动机

PR body 指出该崩溃可从默认配置触达: 在 2 张 GPU 上非 CFG 并行模型自动选择 `ulysses_degree=2` 并批量处理两个 CFG 分支, 服务在 warmup 阶段即失败。作者强调这不是冷僻路径, 而是 *Any SP / KV-gather path with a replicated prefix or suffix and batch > 1* 都会命中。

实现拆解

1. 定位问题: `python/sglang/multimodal_gen/runtime/layers/attention/layer.py` 中的 `USPAttention._gather_sharded_sequence` 静态方法, 在裁剪 replicated prefix/suffix 后直接对切片调用 `sequence_model_parallel_all_gather`。由于切片保留了原始行 stride, 当 `batch > 1` 时切片不是连续张量, 而底层 `all_gather_into_tensor` 要求连续输入。
2. 修复: 在 prefix 和 suffix 两个分支的切片操作后分别追加 `.contiguous()`, 将切片物化为连续张量后再进入集合通信; 不复制场景 (无 replicated 部分) 保持原样直接透传, 避免不必要拷贝。
3. 测试配套: 新增 `test/registered/unit/test_sp_gather_sharded_sequence_contiguous.py` (位于 `python/sglang/multimodal_gen/test/unit/` 下), 通过 `_CaptureAllGather` mock 掉真正的 all-gather, 单进程在 CPU 上验证: 批量下 prefix/suffix 分片均连续、batch=1 旧路径不回归、数值保持不变、无复制时张量透传、prefix+suffix 同时使用被拒绝。
4. CI 验证: PR Test 与 PR Test (Extra) 均通过; AMD ROCm 10 CI 失败, 但与本改动无直接关联, 作者未在本 PR 中处理。

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 `_gather_sharded_sequence`): 核心修复文件, `USPAttention._gather_sharded_sequence` 的两个切片分支追加 `.contiguous()`, 解决

batch>1 时非连续分片导致 all-gather 崩溃的问题。

- python/sglang/multimodal_gen/test/unit/test_sp_gather_sharded_sequence_contiguous.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 _CaptureAllGather, TestGatherShardedSequenceContiguous, test_prefix_shard_is_contiguous_when_batched, test_suffix_shard_is_contiguous_when_batched) : 新增回归测试, mock all-gather 后单进程验证批量下前缀 / 后缀分片连续、数值不变、batch=1 旧路径不回归、异常分支被拒绝。

关键符号: _gather_sharded_sequence

关键源码片段

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

核心修复文件, `USPAttention._gather_sharded_sequence` 的两个切片分支追加 `.contiguous()`, 解决 batch>1 时非连续分片导致 all-gather 崩溃的问题。

```
@staticmethod
def _gather_sharded_sequence(
    tensor: torch.Tensor,
    num_replicated_prefix: int = 0,
    num_replicated_suffix: int = 0,
) -> torch.Tensor:
    # 前缀与后缀同时复制会被拒绝, 二者互斥
    if num_replicated_prefix and num_replicated_suffix:
        raise ValueError(
            "Replicated prefix and suffix cannot be used at the same time."
        )

    # 注意: 沿 dim 1 切片会保留原始行 stride, 只有 batch 为 1 时 PyTorch
    # 才因 size-1 维度忽略 stride 而判定为连续; batch > 1 时必须调
    # .contiguous() 物化, 否则序列并行 all_gather 会报
    # "Tensors must be contiguous" 错误。
    if num_replicated_prefix:
        replicated = tensor[:, :num_replicated_prefix]
        sharded = tensor[:, num_replicated_prefix:].contiguous()
        gathered = sequence_model_parallel_all_gather(sharded, dim=1)
        return torch.cat([replicated, gathered], dim=1)

    if num_replicated_suffix:
        replicated = tensor[:, -num_replicated_suffix:]
        sharded = tensor[:, :-num_replicated_suffix].contiguous()
        gathered = sequence_model_parallel_all_gather(sharded, dim=1)
        return torch.cat([gathered, replicated], dim=1)

    # 无复制时原样透传, 避免不必要的拷贝
    return sequence_model_parallel_all_gather(tensor, dim=1)
```

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

新增回归测试，mock all-gather 后单进程验证批量下前缀 / 后缀分片连续、数值不变、batch=1 旧路径不回归、异常分支被拒绝。

```
class _CaptureAllGather:
    """模拟 all-gather 的替身：记录输入张量并返回拼接结果。"""

    def __init__(self):
        self.inputs = []

    def __call__(self, tensor, dim=-1):
        self.inputs.append(tensor)
        return torch.cat([tensor] * _SP, dim=dim)

class TestGatherShardedSequenceContiguous(unittest.TestCase):
    def _run(self, tensor, **kwargs):
        capture = _CaptureAllGather()
        with patch(f"{_LAYER}.sequence_model_parallel_all_gather", capture):
            out = USPAttention._gather_sharded_sequence(tensor, **kwargs)
        self.assertEqual(len(capture.inputs), 1)
        return out, capture.inputs[0]

    def test_prefix_shard_is_contiguous_when_batched(self):
        # 回归：batch=2 时前缀分支的 all-gather 输入必须是连续张量
        tensor = torch.randn(2, 16, 8)
        _, gathered_input = self._run(tensor, num_replicated_prefix=4)
        self.assertTrue(gathered_input.is_contiguous())

    def test_suffix_shard_is_contiguous_when_batched(self):
        # 回归：batch=2 时后缀分支的 all-gather 输入必须是连续张量
        tensor = torch.randn(2, 16, 8)
        _, gathered_input = self._run(tensor, num_replicated_suffix=4)
        self.assertTrue(gathered_input.is_contiguous())

    def test_batch_one_still_contiguous(self):
        # 旧路径 batch=1 恰好连续，必须保持不回退
        tensor = torch.randn(1, 16, 8)
        for kwargs in ({"num_replicated_prefix": 4}, {"num_replicated_suffix": 4}):
            with self.subTest(**kwargs):
                _, gathered_input = self._run(tensor, **kwargs)
                self.assertTrue(gathered_input.is_contiguous())
```

评论区精华

该 PR 无 review 讨论，但 PR body 提供了详尽的根因分析与验证说明。核心结论：

`contiguous()` 在已是连续张量时是 no-op，因此 batch=1 既有路径不受影响；新测试在修复前

对两个分支均失败，证明测试有效。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 性能风险：.contiguous() 在非连续张量上会触发一次显存拷贝。但 batch>1 的路径此前完全崩溃不可用，修复后才有此开销；batch=1 路径下连续张量调用 contiguous() 为 no-op，不影响现有性能。
2. 回归风险：改动仅涉及 USPAttention._gather_sharded_sequence 内部，该函数被 _forward_with_kv_gather 调用，用于 K/V 的序列并行收集。新增测试覆盖了数值保持与形状，回归面小。
3. 兼容性风险：无公共 API 或配置变更，不影响现有用户配置。
4. 残余风险：仓库中可能存在其他类似的「先切片再集合通信」模式未覆盖，但本 PR 仅针对该单一入口修复；测试 mock 了 all-gather，未在真实多 GPU 环境端到端验证。
 - 影响：影响范围：仅影响 diffusion 多模态生成服务的序列并行（SP）路径，在 2 卡及以上、启用 Ulysses 序列并行且存在 replicated prefix/suffix（常见于 CFG 分组）的配置下，此前 warmup 即崩溃，修复后恢复正常。影响程度：高——修复了一个默认配置下的启动崩溃，对 FLUX.2 等模型在 2 卡部署有直接影响；对 batch=1 既有路径无影响。
 - 团队影响：新增 93 行测试代码，明确覆盖了关键回归场景，便于后续维护。
 - 风险标记：核心路径变更，默认配置崩溃修复，多 GPU 依赖未端到端验证，复制开销仅影响新增可用路径

关联脉络

- PR #38584 [Diffusion] Optimize Qwen-Image-Edit attention on Hopper: 同为 diffusion attention 层（runtime/layers/attention）的性能与正确性改动，展示 diffusion 模块注意力后端的持续演进。
- PR #38549 [Diffusion] Return Qwen-Image-Layered outputs and preserve CFG2 rounding: 涉及 diffusion runtime 的 CFG 分支处理与输出正确性，与本 PR 的 CFG 批量路径同属 diffusion 服务稳定性主题。
- PR #38657 [Diffusion] Set an explicit x264 preset for video output: diffusion 模块测试与工具链改进，与本 PR 一样体现对 diffusion 功能完备性和可测试性的关注。