# PR #33353 完整报告

sgl-project/sglang

[diffusion] reject ring parallelism where it would silently miscompute

合并时间：2026-08-04 11:15

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33353

## 执行摘要

- 一句话：diffusion 注意力路径拒绝 ring 并行，防止静默算错
- 推荐动作：值得精读，虽然改动只有 18 行，但它是一个很好的“fail-fast 安全守卫”范例：在特性未完成前显式拒绝而非静默降级。推荐关注三点：get_ring_parallel_world_size() 与 Ulysses all-to-all 的交互边界、UlyssesAttention 与 USPAttention 的职责划分、以及守卫风格的统一（后续应保持同文件相同模式）。由于是基础注意力层，建议复制 py_compile 之外至少加一个微小的单测覆盖，防止守卫被无意删除。

## 功能与动机

PR 在盘点哪些 sglang-diffusion 模型支持跨节点 Ulysses x Ring 序列并行（参照 #33327 MiniMax-H3 模式）时发现三类共享注意力路径会静默算错：USPAttention 的 replicated-prefix/suffix 路径和 replicated-kv-prefix 路径只做 Ulysses all-to-all、从不跨 ring rank 旋转 KV（影响 Cosmos3、flux/flux_2/ernie_image/helios/joy_image/krea2/glm_image/qwen_image/zimage 等）；UlyssesAttention 的 all-to-all 跑在完整序列并行组而非仅 Ulysses 子组上，ring_degree > 1 时会把数据在 ring rank 间乱序混洗。PR 目标是在方案尚未实现前用显式报错替代静默错误输出。

## 实现拆解

1. 守卫点选择：在 python/sglang/multimodal_gen/runtime/layers/attention/layer.py 中新增三处守卫，复用同文件既有风格（参考 USPAttention masked 路径和 __init__ 的 backend 检查）。
2. UlyssesAttention.__init__：在构造最前面检查 get_ring_parallel_world_size() > 1 即抛 NotImplementedError，因为该类 all-to-all 覆盖整个 SP 组、非 ring-aware，示例文本明确提示“改用 USPAttention 替代”。
3. _forward_with_replicated_prefix：在该方法 docstring 之后、取 SP world size 之前插入 ring 检查，拦截 Cosmos3 cross-attention 等 replicated-prefix 场景。
4. _forward_with_replicated_kv_prefix_split：在获取 sp_rank 之前插入 ring 检查，拦截 flux 等模型的 replicated KV prefix 场景。
5. 配套确认：无新增测试文件；作者通过 py_compile 和手工追踪所有调用方，确认任何现有模型 / 配方都不会触发新守卫（均无 ring_degree > 1 配置）。

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 UlyssesAttention.init, USPAttention._forward_with_replicated_prefix, USPAttention._forward_with_replicated_kv_prefix_split）：唯一变更文件，承载三个 ring 并行安全守卫，直接决定 diffusion 模型在 ring 配置下是报错还是静默算错。

关键符号：UlyssesAttention.init, USPAttention._forward_with_replicated_prefix, USPAttention._forward_with_replicated_kv_prefix_split

## 关键源码片段

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

唯一变更文件，承载三个 ring 并行安全守卫，直接决定 diffusion 模型在 ring 配置下是报错还是静默算错。

```
# UlyssesAttention 的 all-to-all 覆盖整个 SP 组，不区分 ulysses/ring 子组，
# 因此 ring_degree > 1 时数据会在 ring rank 间乱序混洗，必须在构造时直接拒绝。
class UlyssesAttention(nn.Module):
    def __init__(
        self,
        num_heads: int,
        head_size: int,
        num_kv_heads: int | None = None,
        softmax_scale: float | None = None,
        causal: bool = False,
        supported_attention_backends: set[AttentionBackendEnum] | None = None,
        prefix: str = "",
        **extra_impl_args,
    ) -> None:
        super().__init__()
        # 守卫 1：UlyssesAttention 尚未实现 ring 感知的 KV 旋转，ring 下会静默算错
        if get_ring_parallel_world_size() > 1:
            raise NotImplementedError(
                "UlyssesAttention's all-to-all spans the combined sequence "
                "parallel group and is not ring-aware; it would silently "
                "shuffle across ring ranks instead of rotating KV within "
                "them. Ring parallelism is not supported for models still "
                "using UlyssesAttention -- use USPAttention instead."
            )
        if softmax_scale is None:
            self.softmax_scale = head_size**-0.5
        else:
            self.softmax_scale = softmax_scale
        # ... 后续初始化 backend 与 attn_impl ...

    def _forward_with_replicated_prefix(
        self,
        q: torch.Tensor,
        k: torch.Tensor,
```

```
    v: torch.Tensor,
    ctx_attn_metadata,
    num_rep: int,
) -> torch.Tensor:
    """Ulysses attention where the first *num_rep* tokens are replicated
    across SP ranks (e.g. text tokens) and should NOT be duplicated by the
    all-to-all.
    ...
    """
    # 守卫 2：replicated-prefix 路径只做 Ulysses all-to-all，没有 ring 旋转
    if get_ring_parallel_world_size() > 1:
        raise NotImplementedError(
            "USPAttention replicated-prefix/suffix path does not support "
            "ring parallelism yet."
        )
    sp_size = get_ulysses_parallel_world_size()
    sp_rank = get_sp_parallel_rank()

    # 拆分 replicated prefix 与 SP-sharded suffix，仅对 suffix 做 all-to-all
    q_rep, q_shard = q[:, :num_rep], q[:, num_rep:]
    k_rep, k_shard = k[:, :num_rep], k[:, num_rep:]
    v_rep, v_shard = v[:, :num_rep], v[:, num_rep:]
    # ... 后续正常执行 Ulysses attention ...

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 avoids materializing full K/V before Ulysses all-to-all"""
    # 守卫 3：replicated-kv-prefix 路径同样缺少 ring 感知的 KV 旋转
    if get_ring_parallel_world_size() > 1:
        raise NotImplementedError(
            "USPAttention replicated-kv-prefix path does not support "
            "ring parallelism yet."
        )
    sp_rank = get_sp_parallel_rank()
    # ... 后续正常执行 all-to-all 与局部 head shard 切片 ...
```

## 评论区精华

PR 无人工 review 评论；仅有 gemini-code-assist 的自动通知（说明其消费者版本已停止服务）和作者本人的 /tag-and-rerun-ci 指令。核心讨论都在 PR body 内：作者逐路径说明静默算错的具体原因、受影响的模型清单、以及三个守卫分别对应的调用方，并声明通过手工 grep 确认

不会挡住任何现有可用配置。

- ring 并行下静默算错的路径识别与守卫覆盖范围 (design): 作者补上三处 NotImplementedError 守卫，参照同文件既有守卫风格，并手工 grep 确认不阻断任何现有可用配置。无人工 review 分歧。

## 风险与影响

- 风险:
  1. 遗漏调用路径风险：守卫只覆盖已识别的三条路径；若未来新增路径也复用 Ulysses all-to-all 而不做 ring 旋转，仍可能静默算错。删除或绕过库仍有风险，但守卫应尽快补齐。
  2. 误伤风险：守卫是全局 get_ring_parallel_world_size() > 1 检查，若未来某模型在非 ring 场景下也走这些路径，可能直接抛错——不过作者已手工核实当前配置均不会触发。
  3. 测试缺口：无新增测试覆盖这些守卫行为，后续如果有人在 ring 模式下启用这些模型，可能在 CI 中直接失败而非定向测试保护。
  4. 依赖版本 / 行为变化：get_ring_parallel_world_size 返回值可能随并行配置实现变化，若该函数本身有边界行为需关注。 - 影响：影响范围集中在 sglang-diffusion 模块：Cosmos3、flux/flux_2/ernie_image/helios/joy_image/krea2/glm_image/qwen_image/zimage、HunyuanVideo、Wan 等在 --ring-degree > 1 时将立即报错，避免静默产出错误注意力结果；对当前默认配置和现有 recipe 完全兼容。对团队而言，这是跨节点 Ulysses x Ring 方案落地的安全前置，防止后续实验配置误用；对用户而言，误配时从“看似成功但结果错误”变为“立即明确报错”，可调试性显著提升。 - 风险标记：核心注意力路径变更，缺少测试覆盖，守卫覆盖范围有限

## 关联脉络

- PR #33327 [diffusion] MiniMax-H3 cross-node Ulysses x Ring sequence parallelism: PR 明确说明本轮改动起源于对 #33327 的盘点：哪些模型已具备跨节点 Ulysses x Ring 能力，从而发现这三条路径会静默算错。
- PR #33453 [diffusion] Restrict request-level quality to two validated tiers: 同期 diffusion 模块的配置收敛 PR，同为对未验证路径做显式拒绝而非静默放行的维护风格。