# PR #23221 完整报告

sgl-project/sglang

Optimize LTX2 feed-forward tensor parallelism

合并时间：2026-04-21 16:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23221

## 执行摘要

- 一句话：优化 LTX2 前馈网络张量并行，消除大尺寸 AllGather 通信提升推理速度。
- 推荐动作：该 PR 值得精读，重点关注张量并行中激活分片保持的设计决策，以及如何通过 ColumnParallelLinear(gather_output=False) 和 RowParallelLinear(input_is_parallel=True) 的组合消除大尺寸 AllGather。同时可学习其完整的性能验证方法，包括基准测试、内核分析和视觉质量检查。

## 功能与动机

原始实现在张量并行（TP）下，前馈网络的中间激活会在 GELU 激活前通过 AllGather 聚合到所有 TP rank，产生大量通信开销。PR body 明确指出“The old path gathered the expanded FFN hidden state across TP ranks before GELU and the output projection”，优化目标是“removes the large FFN AllGather path while preserving the checkpoint layout”。

## 实现拆解

1. 修改前馈网络初始化配置：在 python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py 的 LTX2FeedForward.__init__ 中，将 self.proj_in 的 gather_output 参数从 True 改为 False，使投影输出保持分片状态；将 self.proj_out 从 ColumnParallelLinear 改为 RowParallelLinear，并设置 input_is_parallel=True 以接受分片输入。
2. 保持前向传播接口不变：forward 方法签名和调用方式未变，仅底层并行策略改变，确保模型输出维度和数值范围与原始实现一致。
3. 验证与基准测试配套：PR 提供了完整的基准测试命令、性能对比表格（包括总请求时间、各阶段耗时）、Nsight Systems 内核分析（显示 AllGather 时间从 12.2% 降至 5.4%）和输出视频视觉检查，但未包含代码变更的直接单元测试。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py（模块 扩散模型；类别 source；类型 core-logic；符号 LTX2FeedForward.init, LTX2FeedForward.forward）：唯一修改的源码文件，包含 LTX2FeedForward 类的张量并行策略调整，直接影响模型推理性能。

关键符号：LTX2FeedForward.init, LTX2FeedForward.forward

## 关键源码片段

### python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py

唯一修改的源码文件，包含 LTX2FeedForward 类的张量并行策略调整，直接影响模型推理性能。

```
class LTX2FeedForward(nn.Module):
    def __init__(
        self,
        dim: int,
        dim_out: int | None = None,
        mult: int = 4,
        quant_config: QuantizationConfig | None = None,
    ) -> None:
        super().__init__()
        if dim_out is None:
            dim_out = dim
        inner_dim = int(dim * mult)

        # 关键变更 1：设置 gather_output=False，使投影输出保持分片状态，避免 AllGather
        self.proj_in = ColumnParallelLinear(
            dim, inner_dim, bias=True, gather_output=False, quant_config=quant_config
        )
        self.act = nn.GELU(approximate="tanh")
        # 关键变更 2：改为 RowParallelLinear，并设置 input_is_parallel=True 以接受分片输入
        self.proj_out = RowParallelLinear(
            inner_dim,
            dim_out,
            bias=True,
            input_is_parallel=True,
            quant_config=quant_config,
        )

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        x, _ = self.proj_in(x) # 输出为分片状态
        x = self.act(x) # GELU 在分片激活上应用
        x, _ = self.proj_out(x) # 行并行投影将分片输入还原为完整隐藏层大小
        return x
```

## 评论区精华

review 讨论较少，gemini-code-assist[bot] 的评论总结了变更要点："updates proj_in to disable output gathering and changes proj_out to a RowParallelLinear layer"，并提到"A new unit test using AST parsing has been added to verify these configurations"，但实际提交中未见测试文件变更，可能评论有误。mickqian 仅批准未发表具体意见。

- 变更总结与测试提及 (other): 评论可能误报测试添加，实际变更仅涉及源码调整。

# 风险与影响

- 风险：
  1. 数值精度风险：由于通信模式从 AllGather+ColumnParallel 改为 RowParallel，浮点累加顺序可能变化，导致输出微小差异。PR body 已通过 PSNR/SSIM 指标验证差异在重复运行波动范围内（主运行间 PSNR 23.14，优化后与主运行间 PSNR 23.74）。
  2. 兼容性风险：仅修改 LTX2FeedForward 类，不影响其他模型或接口，但需确保所有使用该类的场景（如不同 TP size、量化配置）均能正确处理新的并行策略。
  3. 性能回归风险：AllReduce 通信时间增加（bf16 AllReduce 从 24.2% 升至 34.9%），但总体通信开销减少，实测性能提升，风险较低。
- 影响：
  1. 用户影响：LTX2 模型用户无需任何配置变更即可获得推理加速，去噪阶段平均提速 3.5%，精炼阶段提速 26.1%，总请求时间减少约 6%。
  2. 系统影响：减少 AllGather 通信量，降低 GPU 间带宽压力，可能改善多节点扩展性；增加 AllReduce 操作，但整体通信开销下降。
  3. 团队影响：为扩散模型张量并行优化提供了可复用的模式（保持激活分片 + 行并行输出），后续类似模块可参考此设计。 - 风险标记：数值精度微小变化，缺少直接单元测试

# 关联脉络

- PR #20816 [Diffusion][CPU] Init CPU platform support for SGLang Diffusion: 同属 diffusion 模块的优化，涉及多模态生成运行时，但本 PR 聚焦 GPU 张量并行通信优化。