PR #50685 完整报告

vllm-project/vllm

[Bugfix][Refactor] Keep Qwen3Next layer boundaries sequence parallel

合并时间：2026-08-14 11:58

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/50685

执行摘要

- 一句话：修复 Qwen3Next EP+SP 解码损坏，改为显式 SP 布局契约
- 推荐动作：值得精读。核心价值在于「形状推断歧义 → 显式布局契约」的设计决策：当 TP 维度下形状无法区分全量与分片时，放弃运行时推断、在模型入口 / 出口与每层边界固定布局，是比继续补丁形状判断更稳健的方案。建议关注三点：_should_use_sequence_parallel 的判定条件（全 MoE 才启用契约）、入口 shard 与出口 gather 的对称性（含 aux hidden state 的合并 gather）、MTP 绕过共享 forward 时的契约适配方式。

功能与动机

Issue #50681 报告 Qwen3.6-35B-A3B（TP2/DP2/EP4，8×A800，PyTorch 2.13 / CUDA 12.9）在仅开启 EP 与 MoE SP 时产生损坏输出（如 To **E****E****E******** **Quant2**Quant2020**）。PR body 给出的根因是：Qwen3NextModel 与 Qwen3NextDecoderLayer 通过 hidden_states.shape[0] != full_num_tokens 推断当前是否为序列并行分片，但在 TP2 单 token decode 时全量输入与每个 padded 本地 shard 都只有 1 行，shape 无法区分 attention 之前是否必须 all-gather。修复目标是消除运行时布局推断，改为模型整体固定契约，并沿用 Kimi K3 的既有结构。

实现拆解

本 PR 将 Qwen3Next 的序列并行从「逐层形状推断」改为「模型级固定契约」，分四步落地：

1. 新增模型级 SP 判定谓词：在 vllm/model_executor/models/qwen3_next.py 中新增 _should_use_sequence_parallel(vllm_config)，统一从 parallel_config.use_sequence_parallel_moe、pipeline_parallel_size == 1、num_experts > 0、mlp_only_layers 为空、decoder_sparse_step == 1 推导是否启用契约。Qwen3NextDecoderLayer.__init__ 与 InternS2MobiusDecoderLayer.__init__ 中的 use_attn_reduce_scatter_for_moe 改为直接由该谓词赋值，删除了原先耦合 is_moe_layer 的逐层判定。
2. 删除形状推断，固定层边界布局：Qwen3NextDecoderLayer.forward 与 InternS2MobiusDecoderLayer.forward 删除 input_is_sequence_parallel = use_attn_reduce_scatter_for_moe and residual is not None and hidden_states.shape[0] != full_num_tokens 推断，改为只要开启契约就在 attention 前无条件 tensor_model_parallel_all_gather、输出后 reduce_scatter；同时删除

sequence_parallel_chunk(residual) 的兜底分支，层输出恒为序列并行分片。

3. 模型入口 / 出口对称处理：Qwen3NextModel.forward 与 InternS2MobiusModel.forward 新增 use_sequence_parallel property（读取 self.layers[self.start_layer].use_attn_reduce_scatter_for_moe），入口处 sequence_parallel_chunk(hidden_states) 只 shard 一次，删除模型循环中基于 shape 的 _all_gather_hidden_and_residual 调用；出口处在 norm 后统一 all-gather 一次，若存在 aux hidden state 则先拼接再 gather、再按 hidden_size 切回。_all_gather_hidden_and_residual 辅助函数整体删除。
4. MTP 直接调用路径适配：vllm/model_executor/models/qwen3_next_mtp.py 的 Qwen3NextMultiTokenPredictor.forward 与 qwen3_5_mtp.py 的 Qwen3_5MoeMTP.forward 改为：直接调用 decoder layer 前先 sequence_parallel_chunk（并带 hidden_states.shape[0] == positions.shape[-1] 与 residual is None 断言），norm 之后 tensor_model_parallel_all_gather 并截断到 positions.shape[-1]，以对齐主模型的固定契约。
5. 测试与验证配套：本次没有新增自动化测试文件，采用端到端验证。作者用 Qwen/Qwen3.6-35B-A3B（DP2+TP2+EP4+eager+greedy）复现了修复前后的输出差异；评审期间 ZJY0516 补充了 4×GB200 上 GSM8K 500 样本、concurrency 64 的四组配置（compile/eager × MTP K=2）评测，全部 0/500 请求错误且 eager MTP 与非 MTP 输出逐字节一致。

关键文件：

- vllm/model_executor/models/qwen3_next.py（模块 模型层；类别 source；类型 data-contract；符号 _should_use_sequence_parallel, Qwen3NextDecoderLayer.forward, Qwen3NextModel.use_sequence_parallel, Qwen3NextModel.forward）：本 PR 的核心变更文件。新增 _should_use_sequence_parallel 模型级 SP 判定谓词，重写 Qwen3NextDecoderLayer.forward 与 Qwen3NextModel.forward：删除基于 shape 的布局推断与 _all_gather_hidden_and_residual 兜底，改为入口 shard 一次、层边界保持 SP、出口 gather 一次的固定契约，是数据契约重构的主战场。
- vllm/model_executor/models/interns2_mobius.py（模块 模型层；类别 source；类型 data-contract；符号 InternS2MobiusDecoderLayer.forward, InternS2MobiusModel.use_sequence_parallel, InternS2MobiusModel.forward）：Qwen3Next 之外第二个采用相同固定契约的模型。删除对 _all_gather_hidden_and_residual 的导入与模型循环中的形状推断分支，新增 use_sequence_parallel property 与入口 shard/ 出口 gather 的对称处理，属于契约的跨模型推广。
- vllm/model_executor/models/qwen3_next_mtp.py（模块 模型层；类别 source；类型 data-contract；符号 Qwen3NextMultiTokenPredictor.forward）：MTP 实现直接调用 decoder layer、绕过共享 model forward，因此必须在直接调用前手动 shard、norm 后 gather，并新增 shape/None 断言，保证 torch.compile 下符号形状可解析。属于对主模型新契约的必要适配。
- vllm/model_executor/models/qwen3_5_mtp.py（模块 模型层；类别 source；类型 data-contract；符号 Qwen3_5MoeMTP.forward）：与 qwen3_next_mtp.py 完全相同的

契约适配，证明 Qwen3.5 MoE 共享实现下 MTP 路径同样需要入口 / 出口对齐，改动虽小但属于同一数据契约的闭环。

关键符号：_should_use_sequence_parallel, Qwen3NextDecoderLayer.forward, Qwen3NextModel.use_sequence_parallel, Qwen3NextModel.forward, InternS2MobiusModel.forward, Qwen3NextMultiTokenPredictor.forward, Qwen3_5MoeMTP.forward

关键源码片段

vllm/model_executor/models/qwen3_next_mtp.py

MTP 实现直接调用 decoder layer、绕过共享 model forward，因此必须在直接调用前手动 shard、norm 后 gather，并新增 shape/None 断言，保证 torch.compile 下符号形状可解析。属于对主模型新契约的必要适配。

```
current_step_idx = spec_step_idx % self.num_mtp_layers
mtp_layer = self.layers[current_step_idx]
# MTP 直接调用 decoder layer，绕过共享 model forward，
# 因此必须先手动 shard，满足 decoder 的固定 SP 契约；
# assert 同时确保输入确实是完整 token 序列（未重复分片）。
if mtp_layer.use_attn_reduce_scatter_for_moe:
    assert hidden_states.shape[0] == positions.shape[-1]
    hidden_states = sequence_parallel_chunk(hidden_states)
    assert residual is None
hidden_states, residual = mtp_layer(
    positions=positions,
    hidden_states=hidden_states,
    residual=residual,
)

if not get_pp_group().is_last_rank:
    return IntermediateTensors(
        {"hidden_states": hidden_states, "residual": residual}
    )

hidden_states, _ = self.norm(hidden_states, residual)
# MTP norm 之后再 gather 一次，把 SP 分片还原为完整序列，
# 与主模型出口处的行为保持一致。
if mtp_layer.use_attn_reduce_scatter_for_moe:
    hidden_states = tensor_model_parallel_all_gather(hidden_states, 0)
    hidden_states = hidden_states[: positions.shape[-1]]
return hidden_states
```

评论区精华

评审中有三处关键讨论：

1. SP 标志命名与计算位置上移：gcanlin 建议把所有 use_attn_reduce_scatter_for_moe 更名为 use_sequence_parallel，并在 Qwen3NextModel.__init__ 中调用

_use_model_wide_sequence_parallel(vllm_config) 计算一次，而不是每个 decoder layer 各算一次；还指出既然 model 类已有该属性，use_sequence_parallel property 可以去掉。ZJY0516 回应「Then we need to modify every model class, qwen3 next and qwen 3.5」，表明改动面顾虑。合入版本保留了「每层计算 + model 通过 property 读取首层标志」的 折中，重命名未落地。

2. 同类缺陷的模式性推广：thegoldenflow 在 issue #50681 中指出，同样的形状推断逻辑在 deepseek_v2.py 中存在一份独立的更老拷贝（decoder 层入口检查、非 SP 层 all-gather 兜底、aux hidden state gather），早于 #47006，并已另开 #50691 承接 DeepSeek 侧的对应修复，说明这是模式性缺陷而非 Qwen3Next 独有。
3. 端到端评测充分性：ZJY0516 补充了 GSM8K 500 samples 的评测表（compile/eager × MTP K=2 四组），强调当时是用真实 Qwen3.6 MTP 验证作用域：main 上现行 MTP 路径本就输出正常，因此 MTP 文件改动仅是契约适配而非隐藏 bug 修复。

- SP 标志的命名与计算位置上移 (design)：未采纳上移初始化与重命名；合入版本保留「每层计算 use_attn_reduce_scatter_for_moe + model 通过 property 读取首层标志」的折中方案。
- deepseek_v2.py 存在同款形状推断缺陷 (correctness)：本 PR 范围限定 Qwen3Next；DeepSeek 侧的同类缺陷由 #50691 跟进，说明该问题是模式性缺陷。
- 端到端评测覆盖（GSM8K 四组配置） (testing)：四组配置均 0/500 请求错误，修复后输出连贯，且 eager MTP 与非 MTP 输出逐字节一致，验证充分。

风险与影响

- 风险：
  - 混合 dense/MoE 配置行为变化：_should_use_sequence_parallel 对 mlp_only_layers 非空或 decoder_sparse_step != 1 的配置恒返回 False，意味着这些配置的 MoE 层不再启用 reduce-scatter 序列并行。PR body 声称「保留 layer-local MoE 路径」，但合入代码实际上是整体回退到非 SP 布局，属主动取舍，但若有用户依赖混合配置的 MoE SP 加速，会感知性能回退。
  - 无新增自动化测试：4 个文件全为源码改动，无对应单测，布局契约若被后续重构破坏，缺少回归兜底。
  - MTP 断言与 torch.compile 动态形状：qwen3_next_mtp.py 与 qwen3_5_mtp.py 新增 assert hidden_states.shape[0] == positions.shape[-1]，eager 下若上游传入分片会直接断言失败；torch.compile 下依赖显式 token 维度契约让 Dynamo 可编译符号化形状，ZJY0516 的评测覆盖了 compile 配置但未覆盖 PP>1 与变量 batch 上限外场景。
  - 语义命名漂移：use_attn_reduce_scatter_for_moe 已从「本层注意力后是否做 reduce-scatter」漂移为「模型整体是否启用 SP 契约」，后续维护者可能误读。
  - 影响：影响范围集中在 vllm/model_executor/models 下的 Qwen3Next、Qwen3.5 MoE、InternS2Mobius 三个模型族及其 MTP 分支。对用户而言，修复了 EP + MoE SP 组合下单 token decode 的输出损坏这一正确性缺陷；未启用 use_sequence_parallel_moe 或非全 MoE 配置的模型行为与修复前基本一致（后者 MoE SP 优化被禁用）。对团队而言，确立了「模型级固定 SP 契约 + MTP 直接调用适配」的实现范式，参照 Kimi K3，后续新模型（含 DeepSeek 系 #50691）可直接复用该模式；同时删除了

_all_gather_hidden_and_residual 公共辅助函数，其他引用方需注意。

- 风险标记：核心路径变更，缺少测试覆盖，多模型适配，模型布局契约变更

关联脉络

- 暂无明显关联 PR