PR #28516 完整报告

sgl-project/sglang

[NPU] Add MTP support for GLM-4.7-Flash

合并时间：2026-06-18 17:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28516

执行摘要

- 一句话：为 GLM-4.7-Flash 添加 MTP 支持
- 推荐动作：建议合并前处理 torch.zeros 的修改以提升数值稳定性，并补充基本的 MTP 精度测试。此 PR 值得关注，因其展示了在 NPU 上为 MoE 模型适配 MTP 的典型模式：通过 padding 对齐算子约束、绕过 init 时手动配置关键属性。

功能与动机

为 GLM-4.7-Flash 模型启用 MTP 功能，提升推理效率。PR body 中明确说明："Add MTP support for GLM-4.7-Flash"，并指出需要修改两个关键点：1）给 Glm4MoeLiteDecoderLayer 补充缺失属性；2）修改 ascend_backend.py 的 forward_mtp 函数，将 Q 头数 padding 到 2 的幂以适配 FIA 算子。

实现拆解

1. Ascend 注意力后端 Q 头 padding：在 python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py 的 forward_mtp 方法中，当 self.q_head_num_padding > self.tp_q_head_num 时，对 q_nope 和 q_rope 在 head 维度 padding 到 self.q_head_num_padding，使用 torch.empty 创建 padding 张量并 cat 拼接，传入 FIA 算子的 num_heads 参数改为 self.q_head_num_padding，最后切取前 layer.tp_q_head_num 个 head 的输出。
2. 模型层配置修正：在 python/sglang/srt/models/glm4_moe_lite.py 的 Glm4MoeLiteDecoderLayer.__init__ 中添加 config.moe_layer_freq = 1，因为 MTP 启用时模型入口切换至 Glm4MoeLiteModelNextN，会绕过 Glm4MoeLiteForCausalLM.__init__，导致该属性未初始化。
3. 测试配套：未包含测试文件变更。PR body 中未提供精度 / 速度测试结果。

关键文件：

- python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py（模块 算子适配；类别 source；类型 core-logic）：核心实现：在 forward_mtp 中对 Q 头进行 padding 以适配 FIA 算子对 2 的幂次头数的要求。修改了注意力计算的关键路径。
- python/sglang/srt/models/glm4_moe_lite.py（模块 模型定义；类别 source；类型 data-contract）：配置修复：在 Glm4MoeLiteDecoderLayer 中设置 moe_layer_freq=1，确保 MTP 路径下该属性正确初始化。

关键符号：forward_mtp

关键源码片段

python/sglang/srt/hardware_backend/npu/attention/ascend_backend.py

核心实现：在 forward_mtp 中对 Q 头进行 padding 以适配 FIA 算子对 2 的幂次头数的要求。修改了注意力计算的关键路径。

```
def forward_mtp(self, q, k, v, layer, forward_batch, save_kv_cache=True, q_rope=None, k_rope=None, sinks=None):
    # ... 前面的 reshape 和前处理逻辑 ...
    q_nope = q.view(-1, layer.tp_q_head_num, self.kv_lora_rank).contiguous()
    q_rope = q_rope.view(-1, layer.tp_q_head_num, self.qk_rope_head_dim)

    if (
        self.q_head_num_padding is not None
        and self.q_head_num_padding > self.tp_q_head_num
    ):
        # 对 Q 头进行 padding，使得 head 数为 2 的幂，满足 FIA 算子要求
        nope_padding = torch.empty(
            [
                q_nope.shape[0],
                self.q_head_num_padding - self.tp_q_head_num,
                self.kv_lora_rank,
            ],
            dtype=(
                self.model_dtype
                if self.model_dtype is not None
                else torch.bfloat16
            ),
            device=q_nope.device,
        )
        rope_padding = torch.empty(
            [
                q_rope.shape[0],
                self.q_head_num_padding - self.tp_q_head_num,
                self.qk_rope_head_dim,
            ],
            dtype=(
                self.model_dtype
                if self.model_dtype is not None
                else torch.bfloat16
            ),
            device=q_rope.device,
        )
        q_nope = torch.cat([q_nope, nope_padding], dim=1).contiguous()
        q_rope = torch.cat([q_rope, rope_padding], dim=1).contiguous()

    # 调用 FIA 算子，传递 padding 后的 num_heads
```

```
    workspace = torch_npu._npu_fused_infer_attention_score_get_max_workspace(
        q_nope, c_kv_cache, c_kv_cache,
        query_rope=q_rope, key_rope=k_rope_cache,
        num_heads=self.q_head_num_padding,
        num_key_value_heads=layer.tp_k_head_num,
        input_layout="TND", scale=layer.scaling,
        antiquant_mode=0, antiquant_scale=None,
        block_table=self.forward_metadata.block_tables,
        block_size=self.page_size, sparse_mode=3,
        atten_mask=self.mtp_mask,
        actual_seq_lengths=actual_seq_lengths,
        actual_seq_lengths_kv=actual_seq_lengths_kv,
    )
    attn_output = torch.empty_like(q_nope, dtype=q.dtype, device=q.device)
    softmax_lse = torch.empty(1, dtype=q.dtype, device=q.device)
    torch_npu.npu_fused_infer_attention_score.out(
        q_nope, c_kv_cache, c_kv_cache,
        query_rope=q_rope, key_rope=k_rope_cache,
        num_heads=self.q_head_num_padding,
        num_key_value_heads=layer.tp_k_head_num,
        input_layout="TND", scale=layer.scaling,
        antiquant_mode=0, antiquant_scale=None,
        block_table=self.forward_metadata.block_tables,
        block_size=self.page_size, sparse_mode=3,
        atten_mask=self.mtp_mask,
        actual_seq_lengths=actual_seq_lengths,
        actual_seq_lengths_kv=actual_seq_lengths_kv,
        workspace=workspace,
        out=[attn_output, softmax_lse],
    )
    # 截取有效 head 的输出
    attn_output = attn_output[:, : layer.tp_q_head_num, :]
    attn_output = attn_output.view(-1, layer.tp_q_head_num * layer.v_head_dim)
    # ... 后续 padding 处理 ...
```

python/sglang/srt/models/glm4_moe_lite.py

配置修复：在 Glm4MoeLiteDecoderLayer 中设置 moe_layer_freq=1，确保 MTP 路径下该属性正确初始化。

```
class Glm4MoeLiteDecoderLayer(nn.Module):
    def __init__(
        self,
        config: PretrainedConfig,
        layer_id: int,
        quant_config: Optional[QuantizationConfig] = None,
        is_nextn: bool = False,
        prefix: str = "",
        alt_stream: Optional[torch.cuda.Stream] = None,
    ) -> None:
```

```
super().__init__()
# Required for MTP: Glm4MoeLiteModelNextN bypasses Glm4MoeLiteForCausalLM.__init__
config.moe_layer_freq = 1
self.hidden_size = config.hidden_size
# ... 后续初始化 ...
```

评论区精华

1. padding 张量初始化争议：Gemini Code Assist 建议将 torch.empty 改为 torch.zeros 以避免未初始化数据导致 NaN 或非确定性行为，并指出条件中误用 layer.tp_q_head_num 而非 self.tp_q_head_num（实际 diff 显示已使用 self.tp_q_head_num）。作者未回复此评论。
2. config.moe_layer_freq=1 冗余性质疑：Gemini Code Assist 认为该赋值冗余，因为 Glm4MoeLiteForCausalLM.__init__ 已设置此值。作者解释 MTP 路径绕过该 init，因此必须在此设置。Hexq0210 要求添加参数注释，作者已补充注释。

- torch.empty 数值稳定性 (correctness): 作者未回应此评论，未修改。
- config.moe_layer_freq 冗余性 (design): 作者添加了注释澄清原因，评论 resolved。

风险与影响

- 风险：
  1. 数值稳定性风险：使用 torch.empty 初始化 padding 张量可能引入未初始化数据，导致 FIA 算子内部出现 NaN 或 inf，影响推理正确性（参见评论）。建议改用 torch.zeros 消除此风险。
  2. 回归风险：修改 Glm4MoeLiteDecoderLayer.__init__ 会修改共享的 config 对象，可能影响非 MTP 路径，尽管当前 moe_layer_freq 已为 1，但若未来更改默认值则可能产生副作用。
  3. 缺少测试覆盖：无单元测试或集成测试验证 MTP 功能在 GLM-4.7-Flash 上的正确性，增加回归隐患。 - 影响：影响范围：仅限于 Ascend NPU 上运行 GLM-4.7-Flash 模型并启用 MTP 的场景。影响程度中等，因为涉及核心注意力计算路径的修改，但改动量小（+38/-2）。团队需确保 NPU 上的 MTP CI 覆盖此模型。 - 风险标记：使用 uninitialized tensor, 缺少测试覆盖

关联脉络

- PR #28410 [Bugfix] Fix MTP acceptance regression on plan stream by moving int64 cast before plan stream context: 同为 MTP 相关修复，修改了 base_spec_worker.py 等文件，与本 PR 的 MTP 功能有共同上下文。