PR #42787 完整报告

vllm-project/vllm

[MM] Enable FlashInfer metadata support for Qwen2.5-VL vision attention

合并时间：2026-05-24 00:08

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42787

执行摘要

- 一句话：为 Qwen2.5-VL 视觉注意力启用 FlashInfer 元数据支持
- 推荐动作：值得精读，尤其是涉及多模态模型注意力后端切换的开发者。设计上通过 Optional 参数和 padding size 区分不同后端，方法可以借鉴。

功能与动机

使用 FlashInfer 作为 MM encoder 注意力后端时，会触发 assert sequence_lengths is not None 的断言错误（参见 PR body）；为了支持 FlashInfer，需要传递序列长度元数据。

实现拆解

1. 修改注意力前向方法签名（Qwen2_5_VisionAttention.forward 和 Qwen2_5_VisionBlock.forward），将 sequence_lengths 参数的类型从 torch.Tensor 改为 torch.Tensor | None，并添加默认值 None。
2. 在 VisionTransformer.__init__中，根据是否启用 FP8 计算 padded hidden size（通过调用 get_fp8_padded_hidden_size），用于区分 attention 后端是否需要进行元数据计算。
3. 在 VisionTransformer.prepare_encoder_metadata 中，将 cu_seqlens 转换为 numpy 数组，并通过 MMEncoderAttention.maybe_compute_seq_lens 获取每个序列的真实长度（sequence_lengths_full/sequence_lengths_window），这些元数据后续传递给 attention 层。
4. 在 Qwen2_5_VisionBlock.forward 中，将 sequence_lengths 参数从硬编码的 None 改为从调用方传入，并将该参数添加到 torch.compile 的动态参数集，避免 dynamo 错误。
5. 此变更未涉及测试文件，但通过手动 benchmark 验证了性能。

关键文件：

- vllm/model_executor/models/qwen2_5_vl.py（模块 VL 模型；类别 source；类型 data-contract；符号 Qwen2_5_VisionAttention.forward, Qwen2_5_VisionBlock.forward, Qwen2_5_VisionTransformer.init, Qwen2_5_VisionTransformer.prepare_encoder_metadata）：唯一变更文件，核心修复

关键符号：Qwen2_5_VisionAttention.forward, Qwen2_5_VisionBlock.forward, Qwen2_5_VisionTransformer.prepare_encoder_metadata, Qwen2_5_VisionTransformer.init

关键源码片段

vllm/model_executor/models/qwen2_5_vl.py

唯一变更文件，核心修复

```
def forward(
    self,
    x: torch.Tensor,
    cu_seqlens: torch.Tensor,
    rotary_pos_emb_cos: torch.Tensor,
    rotary_pos_emb_sin: torch.Tensor,
    max_seqlen: torch.Tensor, # 仅用于 Flash Attention
    # 仅用于 FlashInfer CuDNN 后端，允许为 None 以兼容其他后端。
    sequence_lengths: torch.Tensor | None,
) -> torch.Tensor:
    # [s, b, c] --> [s, b, head * 3 * head_dim]
    x, _ = self.qkv(x)
    seq_len, batch_size, _ = x.shape

    qkv = einops.rearrange(
        x,
        "s b (three head head_dim) -> b s three head head_dim",
        three=3,
        head=self.num_attention_heads_per_partition,
    )

    # 旋转位置编码：qk 统一旋转后再拆分
    if rotary_pos_emb_cos is not None and rotary_pos_emb_sin is not None:
        qk, v = qkv[:, :, :2], qkv[:, :, 2]
        qk_reshaped = einops.rearrange(
            qk, "b s two head head_dim -> (two b) s head head_dim", two=2
        )
        qk_reshaped = qk_reshaped.contiguous()
        qk_rotated = self.apply_rotary_emb(
            qk_reshaped,
            rotary_pos_emb_cos,
            rotary_pos_emb_sin,
        )
        qk_rotated = qk_rotated.view(
            2,
            batch_size,
            seq_len,
            self.num_attention_heads_per_partition,
            self.hidden_size_per_attention_head,
        )
        q, k = qk_rotated.unbind(dim=0)
    else:
        q, k, v = qkv.unbind(dim=2)
```

```
# 调用 MMEncoderAttention，传入 sequence_lengths（可能为 None）
context_layer = self.attn(
    query=q,
    key=k,
    value=v,
    cu_seqlens=cu_seqlens,
    max_seqlen=max_seqlen,
    sequence_lengths=sequence_lengths,
)

context_layer = einops.rearrange(
    context_layer, "b s h d -> s b (h d)", b=batch_size
).contiguous()

output, _ = self.proj(context_layer)
return output
```

评论区精华

1. gemini-code-assist[bot] 建议显式调用 .cpu() 再 .numpy() 以避免未来 tensor 在 GPU 上时报错，作者已采纳修改。
2. johncalesp 关心 torch compile 与其他注意力后端（FLASH_ATTN/TRITON_ATTN）的兼容性，作者随后在 B200 和 H100 上验证无问题。
3. johncalesp 指出 get_fp8_padded_hidden_size 在未启用 FP8 时返回 None，可能未被 PR 的 benchmark 覆盖；作者解释这是设计意图，用于区分 BF16 和 FP8 布局，且额外测试了 mm-encoder-attn-dtype fp8 也工作正常。

- .cpu() 调用安全性 (correctness): 作者已采纳修改，添加了 .cpu() 调用。
- Torch compile 与其他后端兼容性 (testing): 作者在 B200 和 H100 上测试了 FLASH_INFER, FLASH_ATTN, TRITON_ATTN，均正常工作。
- get_fp8_padded_hidden_size 返回 None 的处理 (design): 作者解释返回 None 是设计意图，用于区分 BF16 (interleaved QKV) 和 FP8 (独立 contiguous) 布局；额外测试了 mm-encoder-attn-dtype fp8 正常。johncalesp 标记可解决。

风险与影响

- 风险：
  1. 后端兼容性风险：sequence_lengths 变为可选后，如果其他后端（如 TRITON_ATTN）意外依赖该参数，可能表现不正确。但代码中其他后端会忽略该参数（因为之前传入 None），风险较低。
  2. Torch compile 动态参数：新增 sequence_lengths 到 dynamic_arg_dims，可能会影响编译缓存，但改动很小。
  3. FP8 条件路径：get_fp8_padded_hidden_size 返回 None 时，后续逻辑是否都正确处理？目前看是设计为无害的 None，但需注意未来扩展。 - 影响：影响范围：使用 Qwen2.5-VL 模型并设置 mm-encoder-attn-backend=FLASHINFER 的用户。影响程度

：修复了崩溃，提升了可用性。对其他后端无负面影响。性能方面，通过 benchmark 显示 token throughput 约 5907 tok/s，合理。 - 风险标记：后端兼容性，Torchcompile 动态参数，FP8 条件路径

关联脉络

- PR #38822 [Attention] Add head_dim=512 support for FlashInfer trtllm attention backend: 都涉及 FlashInfer 注意力后端的功能扩展。
- PR #42546 [ModelOpt] Support Qwen3.5/3.6 VLM quantized prefix mapping: 都是对 Qwen VLM 系列模型的量化与后端支持增强。