PR #25045 完整报告

sgl-project/sglang

Fix FA3 cross-attention batched-decode for per-request varlen encoder

合并时间：2026-05-26 14:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25045

执行摘要

- 一句话：修复 FA3 跨注意力对变长编码器的批处理支持
- 推荐动作：建议精读该 PR，尤其是 init_forward_metadata 中 fancy indexing 的构造方式。这是一个典型的注意力元数据构建 bug 修复，设计简洁，且验证充分，对于理解 SGLang 中跨注意力的批处理实现有很好的参考价值。

功能与动机

FA3 注意后端的跨注意力元数据构建器仅支持 encoder_lens.numel() == 1，在非 CUDA Graph 路径中断言检查，CUDA Graph 重放路径则对所有请求复用 encoder_lens[0]。对于同一批次中混合不同编码器长度的多模态编码器 - 解码器模型，只有第 0 个请求获得正确的跨注意力，其余请求读取错误的 KV 缓存位置，产生乱码。在 bs=128 时，乱码输出比例高达 25-36%。

实现拆解

1. 在 init_forward_metadata 方法中移除 assert encoder_lens.numel() == 1 断言及相关注释。
2. 在 init_forward_metadata 方法中，为自注意力（文本）的 page_table 构建使用基于每个请求的偏移量：通过 fancy indexing 构造 text_col = encoder_lens[i] + arange_text，替换原来使用单一 encoder_max_seq_len_k 全局偏移的方式。
3. 在 init_forward_metadata_replay_cuda_graph 方法中，将 encoder_max_seq_len_k 从 encoder_lens[0] 改为 encoder_lens.max().item()，并将完整的 encoder_lens[:bs] 复制到 int32 和 cu_seqlens 缓冲区（原仅复制 encoder_lens[:1]）。同时调整 page_table 的复制范围以匹配每个请求的实际长度。

关键文件：

- python/sglang/srt/layers/attention/flashattention_backend.py（模块 注意力；类别 source；类型 core-logic）：唯一修改的文件，包含 FA3 注意力后端中跨注意力元数据构建的正确性修复，是修复的核心位置。

关键符号：未识别

关键源码片段

python/sglang/srt/layers/attention/flashattention_backend.py

唯一修改的文件，包含 FA3 注意力后端中跨注意力元数据构建的正确性修复，是修复的核心位置。

```
# python/sglang/srt/layers/attention/flashattention_backend.py # 在 init_forward_metadata 方法中（非 CUDA Graph 路径的跨注意力处理部分） if forward_batch.encoder_lens is not None:     # 移除原有的 assert 断言和注释，现支持变长编码器批处理     metadata.encoder_lens_int32 = forward_batch.encoder_lens.to(torch.int32)     metadata.encoder_cu_seqlens_k = torch.nn.functional.pad( torch.cumsum(metadata.encoder_lens_int32, dim=0, dtype=torch.int32),         (1, 0),     )     metadata.encoder_max_seq_len_k = metadata.encoder_lens_int32.max().item() # 跨注意力 page_table：每行对应一个请求，cache_seqlens 中已     # 记录每行实际编码器长度，超出 encoder_lens[i] 的垃圾数据不会被读取     metadata.encoder_page_table = self.req_to_token_pool.req_to_token[         forward_batch.req_pool_indices, : metadata.encoder_max_seq_len_k     ]     # 自注意力（文本） page_table：文本偏移量基于每个请求的     # encoder_lens[i]，而非单一最大值。使用 fancy index 进行按行偏移     text_max = metadata.max_seq_len_k     arange_text = torch.arange(         text_max, device=forward_batch.req_pool_indices.device     )     # text_col 形状：(bs, max_seq_len_k)，每行从 encoder_lens[i] 开始偏移     text_col = forward_batch.encoder_lens.long().unsqueeze(1) + arange_text.unsqueeze(0) text_row = forward_batch.req_pool_indices.unsqueeze(1).expand(-1, text_max) metadata.page_table = self.req_to_token_pool.req_to_token[         text_row, text_col     ]  # init_forward_metadata_replay_cuda_graph 方法中（CUDA Graph 重放路径） if encoder_lens is not None:     # 支持每请求变长编码器（例如 MossVL 不同图片尺寸） metadata.encoder_max_seq_len_k = int(encoder_lens.max().item())     # 复制完整 batch 的 encoder_lens，而非仅复制第一个     metadata.encoder_lens_int32[:bs].copy_(encoder_lens[:bs].to(torch.int32))     metadata.encoder_cu_seqlens_k[1 : bs + 1].copy_( torch.cumsum(metadata.encoder_lens_int32[:bs], dim=0, dtype=torch.int32)     )     # 限制复制范围为实际请求数 bs     metadata.encoder_page_table[:bs, : metadata.encoder_max_seq_len_k].copy_(         self.req_to_token[req_pool_indices, : metadata.encoder_max_seq_len_k]     )     # 更新常规 page_table 时同样使用每请求偏移   # ...（通过类似 fancy-index gather 实现）
```

评论区精华

该 PR 的 review 评论数量为 0，审核人 mickqian 直接批准，没有留下讨论记录。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更仅影响 FA3 后端的跨注意力路径，且改动逻辑在 PR body 中通过对比测试验证（bs 8~128 下乱码率从 12-36% 降为 0%）。但变更未新增测试，依赖现有的 CI 测试覆盖；此外 fancy indexing 的引入可能带来极其微小的性能开销，但作者在 PR body 中通过速度测试证明无 measurable change。

- 影响：影响仅限于使用 FA3 解码注意力后端、且模型包含跨注意力的多模态编码器 - 解码器模型（如 MossVL）。对于使用 FlashInfer 后端的用户无影响。修复后，这类模型在批处理推理和强化学习 rollout 中的正确性得到保证。
- 风险标记：缺少测试覆盖，核心路径变更

关联脉络

- 暂无明显关联 PR