

PR #22953 完整报告

sgl-project/sglang

[diffusion][bugfix] avoid illegal memory access in qwen image

合并时间: 2026-04-23 12:41

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/22953>

执行摘要

- 一句话: 修复 QwenImage 模型在输入图像过多时 RoPE 文本缓存溢出导致的非法内存访问错误。
- 推荐动作: 该 PR 值得快速浏览, 重点关注 `get_freqs_cis` 方法中的校验逻辑设计。它展示了在 GPU 密集计算前添加防御性校验的最佳实践, 以及如何通过描述性错误信息提升用户体验。对于处理类似缓存溢出问题的开发者有参考价值。

功能与动机

根据 PR body 中的错误报告, 当用户提供大量输入图像 (例如 32 张) 时, `qwen-image-edit` 模型在去噪阶段会因 RoPE 文本缓存溢出而触发 CUDA 非法内存访问错误 (`cudaErrorIllegalAddress`)。错误堆栈指向 `apply_flashinfer_rope_qk_inplace` 函数中的 `positions.to(q.device)` 操作。根本原因是文本序列长度超过了预分配的 RoPE 缓存大小, 导致后续 GPU 内存访问越界。

实现拆解

1. 入口点与校验逻辑: 在 `python/sglang/multimodal_gen/configs/pipeline_configs/qwen_image.py` 的 `get_freqs_cis` 静态方法中, 于生成 flashinfer RoPE 缓存之前, 新增文本序列长度校验。
2. 核心校验步骤: 计算输入文本序列的最大长度 (`max_txt_seq_len`), 获取旋转编码器返回的文本频率缓存长度 (`txt_cache_len`)。若 `max_txt_seq_len > txt_cache_len`, 则计算溢出量并抛出 `ValueError`, 错误信息包含具体数值和解决建议。
3. 影响与后续处理: 该校验在去噪阶段之前执行, 提前阻止非法内存访问, 避免后续 CUDA 内核错误。错误信息明确指导用户调整输入参数 (减少图像数量、缩短提示、降低分辨率), 提升用户体验和调试效率。
4. 测试与配置配套: 本次变更仅修改源码文件, 未包含直接对应的测试文件变更或配置调整。

关键文件:

- `python/sglang/multimodal_gen/configs/pipeline_configs/qwen_image.py` (模块 扩散配置; 类别 `source`; 类型 `core-logic`; 符号 `get_freqs_cis`): 这是唯一修改的文件, 包含修复 RoPE 文本缓存溢出问题的核心校验逻辑。

关键符号: `get_freqs_cis`

关键源码片段

[python/sglang/multimodal_gen/configs/pipeline_configs/qwen_image.py](#)

这是唯一修改的文件，包含修复 RoPE 文本缓存溢出问题的核心校验逻辑。

```
@staticmethod
def get_freqs_cis(img_shapes, txt_seq_lens, rotary_emb, device, dtype):
    # img_shapes: for global entire image
    img_freqs, txt_freqs = rotary_emb(img_shapes, txt_seq_lens, device=device)

    # 新增校验：计算最大文本序列长度，并与缓存长度比较
    max_txt_seq_len = max(txt_seq_lens) if txt_seq_lens else 0
    txt_cache_len = int(txt_freqs.shape[0]) # 获取旋转编码器返回的缓存长度
    if max_txt_seq_len > txt_cache_len:
        overflow = max_txt_seq_len - txt_cache_len
        raise ValueError(
            "QwenImage RoPE text cache overflow before denoising: "
            f"required_txt_seq_len={max_txt_seq_len}, txt_cache_len={txt_cache_len}, "
            f"overflow={overflow}. "
            "Please reduce the number of input images, shorten the prompt, "
            "or lower the requested resolution."
        ) # 抛出描述性错误，指导用户调整参数

    # flashinfer RoPE expects a float32 cos/sin cache concatenated on the last dim
    img_cos_half = img_freqs.real.to(dtype=torch.float32).contiguous()
    img_sin_half = img_freqs.imag.to(dtype=torch.float32).contiguous()
    txt_cos_half = txt_freqs.real.to(dtype=torch.float32).contiguous()
    txt_sin_half = txt_freqs.imag.to(dtype=torch.float32).contiguous()

    img_cos_sin_cache = torch.cat([img_cos_half, img_sin_half], dim=-1)
    txt_cos_sin_cache = torch.cat([txt_cos_half, txt_sin_half], dim=-1)
    return img_cos_sin_cache, txt_cos_sin_cache
```

评论区精华

Review 中仅有 gemini-code-assist[bot] 的自动评论，总结了变更内容为“引入校验以防止 RoPE 文本缓存溢出”，并指出“没有反馈可提供”。维护者 mickqian 直接批准，未引发技术讨论。这表明变更逻辑清晰、风险可控，团队对修复方案达成一致。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 回归风险：新增校验可能引入误报，若 `txt_seq_lens` 计算或 `txt_freqs.shape` 解析有误，可能导致合法请求被错误拒绝。但校验逻辑简单，仅比较两个整数，风险较低。
 2. 性能影响：增加 `max(txt_seq_lens)` 计算和条件判断，对整体推理性能影响可忽略不计。
 3. 兼容性：无 API 或数据格式变更，完全向后兼容。

4. 安全风险：通过提前校验避免内存越界，提升了系统稳定性，降低了因非法内存访问导致的潜在安全漏洞（如数据泄露或崩溃）。

• 影响：

1. 用户影响：修复了特定场景（多图像输入）下的崩溃问题，提升了模型鲁棒性；错误信息更友好，指导用户调整参数。
2. 系统影响：仅影响 QwenImage 扩散模型管道，阻止了缓存溢出导致的 CUDA 错误，避免后续处理中的未定义行为。
3. 团队影响：变更集中且简单，易于理解和维护；未修改核心算法，不影响其他模块。 -
风险标记：缓存溢出风险

关联脉络

- PR #23198 [diffusion] Fix --warmup-resolutions hang with --enable-cfg-parallel: 同属 diffusion 模块的 bugfix，涉及多模态生成管道的调度和预热问题，可对比学习扩散模型中的边缘场景处理。
- PR #22931 [Fix/Kernel] Add JIT rmsnorm_hf kernel to fix transformers backend MMLU accuracy regression : 同为修复 CUDA 内核相关问题的 bugfix，涉及 JIT 内核和准确性回归，展示了不同场景下的内存 / 计算错误处理模式。