PR #40917 完整报告

vllm-project/vllm

[Bugfix][Granite4Vision] Fix deepstack buffer causing decode slowdown in compiled mode

合并时间：2026-04-28 15:43

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40917

执行摘要

- 一句话：修复 Granite4Vision 编译模式下 deepstack 缓存区导致解码性能下降
- 推荐动作：此 PR 是一行关键修复，值得所有使用 Granite4Vision 模型的用户关注。它也是说明 torch.compile 下缓冲区形状影响编译器优化的重要案例，对理解 vLLM 编译管线的性能调优有参考价值。

功能与动机

PR #40282 引入了 Granite4Vision 模型支持，但在 torch.compile 模式下存在性能缺陷：将完整大小的 deepstack 缓冲区传递给编译模型，导致 Inductor 为全部 max_num_batched_tokens（通常 8192）生成计算核，严重降低解码吞吐。作者 artem-spector 在评论中描述症状为“100% SM utilization during single-sequence decode (should be ~5%), 5x lower generation throughput in compiled vs eager mode”。

实现拆解

1. 定位问题根因：在 vllm/model_executor/models/granite4_vision.py 的 forward 方法中，构建 IntermediateTensors 时直接传递了未截取的完整缓冲区 self._ds_buffers[lvl]，其大小为 (max_num_batched_tokens, hidden_dim)。
2. 修复缓冲区截取：在构造 IntermediateTensors 之前，计算 inputs_embeds.size(0) 获取当前实际 token 数 n，然后对每个缓冲区执行 self._ds_buffers[lvl][:n] 切片操作，仅将实际有效数据传入模型。
3. 保持兼容性：切片操作不影响缓冲区零清除逻辑（已使用 buf[:n].zero_()），且对 eager 模式无影响（eager 模式原本就会按实际大小计算）。

关键文件：

- vllm/model_executor/models/granite4_vision.py（模块 模型执行；类别 source；类型 core-logic）：核心修复文件，修改了 forward 方法中 deepstack 缓冲区的切片逻辑，防止完整缓冲区传入编译模型导致内核膨胀。

关键符号：未识别

关键源码片段

vllm/model_executor/models/granite4_vision.py

核心修复文件，修改了 forward 方法中 deepstack 缓冲区的切片逻辑，防止完整缓冲区传入编译模型导致内核膨胀。

```
# 修复前：传入完整缓冲区，导致编译内核处理全部行
if (
    inputs_embeds is not None
    and get_pp_group().is_first_rank
    and self._ds_layer_indices
):
    ds: IntermediateTensors | None = IntermediateTensors(
        {
            f"ds_{llm_layer}": self._ds_buffers[lvl] # 完整缓冲区 (max_num_batched_tokens, hidden)
            for lvl, llm_layer in enumerate(self._ds_layer_indices)
        }
    )
else:
    ds = None

# 修复后：按实际 token 数切片，使编译内核按实际批次大小执行
if (
    inputs_embeds is not None
    and get_pp_group().is_first_rank
    and self._ds_layer_indices
):
    n = inputs_embeds.size(0) # 当前实际 token 数
    ds: IntermediateTensors | None = IntermediateTensors(
        {
            f"ds_{llm_layer}": self._ds_buffers[lvl][:n] # 切片后传入实际数据
            for lvl, llm_layer in enumerate(self._ds_layer_indices)
        }
    )
else:
    ds = None
```

评论区精华

该 PR 的 Review 评论较少，主要由机器人（Claude、Gemini）自动审查且无实质性反馈；维护者 DarkLight1337 直接批准合并。后续 issue 讨论中，作者请求将修复回溯到 v0.20.1 未果（因该 PR 在 v0.20.0 之后才合并），预计包含在 v0.21 中（约 5 月 13-14 日发布）。

- 无法回溯到 v0.20.1 导致已发布版本模型损坏 (other): 维护者 DarkLight1337 解释 v0.20.1 基于 v0.20.0 构建，由于此 PR 未包含在 v0.20.0 中，因此无法回溯到 v0.20.1；预计包含在 v0.21（约 5 月 13-14 日发布）中。

风险与影响

- 风险：本变更仅修改一行代码（加一行、改一行），且逻辑是常见的张量切片操作，风险极低。可能的风险是如果 inputs_embeds 为 None 而进入该分支（理论不会，因外层条件已检查 inputs_embeds is not None），会触发 AttributeError。但鉴于条件检查完备，基本

无隐患。

- 影响：直接修复 Granite4Vision 模型在 torch.compile 模式下的严重性能回归，影响所有使用该模型且启用编译的用户。修复后，decode 阶段的 SM 利用率从 ~100% 降至 ~5%，生成吞吐量提升约 5 倍。不涉及 API 或配置变更，对非编译用户无影响。
- 风险标记：低风险，单行修复

关联脉络

- PR #40282 [Model] Add IBM Granite 4.1 Vision support: 该 PR 引入了 Granite4Vision 模型，同时引入了 deepstack 缓冲区机制。本修复 PR 正是针对该模型在 torch.compile 模式下的性能回退问题，属于同一功能线的延续修复。