PR #40660 完整报告

vllm-project/vllm

[MM][Perf][CG] Support ViT full cudagraphs for mllama4

合并时间：2026-06-12 13:17

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40660

执行摘要

- 一句话：支持 mllama4 视觉编码器全 CUDA 图
- 推荐动作：推荐深入阅读 mllama4.py 中的接口实现，尤其是 get_encoder_cudagraph_budget_range 和 get_encoder_cudagraph_item_specs 的设计逻辑。Review 中关于接口合并和性能建议的讨论值得关注。如果团队计划支持更多多模态模型的 CUDA 图，此 PR 是很好的学习范例。

功能与动机

关联 Issue #38175 指出，多模态大模型的 ViT 编码器前向需要启动大量小 kernel，产生显著的主机端开销。vLLM 已对 decoder (LLM) 部分支持 CUDA 图，但 encoder 仍为 eager 执行。扩展 CUDA 图到 ViT 编码器可以消除每步 kernel 启动开销，实现更低延迟和更高吞吐。PR body 也展示了 benchmark 对比，验证了性能提升。

实现拆解

1. 核心模型接口实现（vllm/model_executor/models/mllama4.py）：让 Llama4ForConditionalGeneration 继承 SupportsEncoderCudaGraph 接口，并实现全部必需方法：get_encoder_cudagraph_config 返回模块配置（modalities、buffer_keys、out_hidden_size）；get_encoder_cudagraph_budget_range 计算最小（每 chunk patches 数）和最大（batch token 与模型长度 min）预算；get_encoder_cudagraph_item_specs 为每个图像基于 chunks 数量生成 EncoderItemSpec；select_encoder_cudagraph_items 根据 indices 子选择 pixel_values 和 patches_per_image；prepare_encoder_cudagraph_capture_inputs 构造 dummy 输入用于 graph 捕获；encode_image_chunks 实际调用视觉模型和投影。同时添加 supports_encoder_cudagraph = True 类属性，并更新 Llama4ImagePatchInputs 中注释以澄清 patches_per_image 语义。
2. CI 测试集成（tests/models/multimodal/generation/test_vit_cudagraph.py）：在 MODEL_CONFIGS 中添加 llama4 条目，使用 dummy 权重和减少的层数以避免 OOM，设置合适的 prompt 和参数，确保自动化测试覆盖。
3. 测试工具适配（tests/models/utils.py）：在 dummy_hf_overrides 中针对 Llama4ForConditionalGeneration 设置 num_experts_per_tok = 1，避免 MoE 配置与默认值不匹配导致的测试问题。

4. 示例更新（examples/generate/multimodal/vision_language_offline.py）：将 "llama4" 加入 MODELS_SUPPORT_VIT_CUDA_GRAPH 列表，使离线运行示例支持该特性。
5. 文档补充（docs/design/cuda_graphs_multimodal.md）：在支持表格中添加 Llama4 行（仅图像），并提供启动命令示例。

关键文件：

- vllm/model_executor/models/mllama4.py（模块 视觉编码器；类别 source；类型 core-logic；符号 get_image_patches_per_chunk, encode_image_chunks, get_encoder_cudagraph_config, get_input_modality）：核心实现文件，添加了 SupportsEncoderCudaGraph 接口的全部方法，改动量最大（+160/-12），是性能优化的主战场。
- tests/models/multimodal/generation/test_vit_cudagraph.py（模块 集成测试；类别 test；类型 test-coverage）：添加 llama4 集成测试配置，确保 CI 覆盖新功能。
- tests/models/utils.py（模块 测试工具；类别 test；类型 test-coverage）：调整 dummy_hf_overrides 以适配 Llama4 的 MoE 配置，保证测试正确性。
- examples/generate/multimodal/vision_language_offline.py（模块 示例脚本；类别 source；类型 core-logic）：将 llama4 加入离线示例的 CUDA 图支持列表，方便用户参考。
- docs/design/cuda_graphs_multimodal.md（模块 设计文档；类别 docs；类型 documentation）：更新文档，记录 Llama4 的支持状态和使用方法。

关键符号：get_image_patches_per_chunk, encode_image_chunks, get_encoder_cudagraph_config, get_input_modality, get_encoder_cudagraph_budget_range, get_encoder_cudagraph_item_specs, select_encoder_cudagraph_items, prepare_encoder_cudagraph_capture_inputs

关键源码片段

vllm/model_executor/models/mllama4.py

核心实现文件，添加了 SupportsEncoderCudaGraph 接口的全部方法，改动量最大（+160/-12），是性能优化的主战场。

```
# 在类定义中新增 SupportsEncoderCudaGraph 混入
class Llama4ForConditionalGeneration(
    ...
    SupportsEncoderCudaGraph,
    ...
):
    supports_encoder_cudagraph = True # 标记支持

    def get_encoder_cudagraph_config(self):
        """返回 encoder CUDA 图配置"""
        from vllm.v1.worker.encoder_cudagraph_defs import EncoderCudaGraphConfig
        return EncoderCudaGraphConfig(
            modalities=["image"],
            buffer_keys=["pixel_values"],
            out_hidden_size=self.config.text_config.hidden_size,
```

```
        )

    def get_encoder_cudagraph_budget_range(self, vllm_config: VllmConfig) -> tuple[int, int]:
        """计算捕获预算范围：最小为每 chunk 的 patches 数，最大受限于 batch token
        上限和模型最大长度"""
        min_budget = self.get_image_patches_per_chunk()
        max_budget = min(
            vllm_config.scheduler_config.max_num_batched_tokens,
            self.vllm_config.model_config.max_model_len,
        )
        return (min_budget, max_budget)

    def get_encoder_cudagraph_item_specs(self, mm_kwargs: dict[str, Any]):
        """为每个图像生成 encoder item spec，基于 chunks 数量"""
        from vllm.v1.worker.encoder_cudagraph_defs import EncoderItemSpec
        patches_per_chunk = self.get_image_patches_per_chunk()
        return [
            EncoderItemSpec(
                input_size=num_chunks, # 输入为 chunk 数
                output_tokens=num_chunks * patches_per_chunk,
            )
            for num_chunks in mm_kwargs["patches_per_image"].tolist()
        ]

    def select_encoder_cudagraph_items(self, mm_kwargs: dict[str, Any], indices: list[int]) ->
    dict[str, Any]:
        """根据选中的 indices 子选择 encoder 输入"""
        pixel_values = mm_kwargs["pixel_values"]
        patches_per_image = mm_kwargs["patches_per_image"]
        if len(indices) == 0:
            return {"pixel_values": pixel_values[:0], "patches_per_image": patches_per_image[:0]}
        # 计算累积 chunk 索引
        cum_chunks = torch.cumsum(patches_per_image, dim=0).tolist()
        cum_chunks = [0] + cum_chunks # 前补 0 便于切片
        selected_pixel_values = torch.cat([
            pixel_values[cum_chunks[i]:cum_chunks[i+1]]
            for i in indices
        ])
        selected_patches = patches_per_image[indices]
        return {"pixel_values": selected_pixel_values, "patches_per_image": selected_patches}
```

评论区精华

- 接口合并提醒：shen-shanshan 指出 get_encoder_cudagraph_budget_range、get_encoder_cudagraph_item_specs、select_encoder_cudagraph_items 三方法已在 PR #41234 中合并为单一方法，本 PR 需同步更新。作者确认将在后续处理。
- 性能优化建议：gemini-code-assist[bot] 建议使用 torch.cumsum 替代循环计算累积 chunks，以及使用 torch.zeros 代替 torch.randn 构造 dummy 输入，以避免潜在的数值

问题。

- 数据并行兼容性：shen-shanshan 询问在 prepare_encoder_cudagraph_capture_inputs 中固定 use_data_parallel=False 的原因，作者通过测试 --mm-encoder-tp-mode data 验证了兼容性。
- CI OOM 风险：shen-shanshan 和 Isotr0py 担心 17B 模型在 CI 中导致 OOM，建议使用 dummy 权重并减少层数，最终测试配置采用了该方法。
- 无关改动清理：Isotr0py 指出 PR 包含不相关的修改（如 test_mllama4.py、test_common.py），要求清理，作者回应已处理。
- 接口合并提醒 (design): 作者回应将在后续处理，并最终适配了新接口。
- 性能优化建议（torch.cumsum）（performance): 建议被采纳，最终代码中使用了 torch.cumsum。
- dummy 输入数值稳定性 (correctness): 建议被采纳，最终使用了 torch.zeros。
- 数据并行兼容性 (correctness): 作者通过实际测试验证了兼容性，确认无需修改。
- CI OOM 风险与测试配置 (testing): 最终测试配置采用 dummy 权重和缩小模型，避免了资源问题。
- 无关改动清理 (other): 作者回应已清理无关改动。

风险与影响

- 风险：
  - CUDA 图捕获失败：若预算计算不准确或模型权重异常，可能导致 capture 失败并退化为 eager，需确保 fallback 路径健壮。
  - 内存占用增加：CUDA 图会预分配缓冲，在长序列或大 batch 时可能额外消耗显存，但预算范围已受 max_num_batched_tokens 限制。
  - 接口耦合风险：SupportsEncoderCudaGraph 接口仍在演进（如 PR #41234 合并为单一方法），本 PR 实现可能需跟随上游更新。
  - 测试资源压力：CI 中使用 dummy 权重和缩减配置，但若未来扩展为全权重测试可能超时或 OOM，需留意硬件限制。
- 影响：
  - 用户影响：使用 Llama-4 并启用 cudagraph_mm_encoder 的用户将获得显著性能提升（请求吞吐量 +22%，TTFT 中位数 -35%）；未启用者不受影响。
  - 系统影响：ViT 编码器变为 CUDA 图执行，降低 CPU kernel 启动开销，提高 GPU 利用率。
  - 团队影响：为后续多模态模型集成 encoder CUDA 图提供了参考模式，但需维护接口兼容性。
  - 风险标记：新接口依赖，CUDA 图捕获失败，CI OOM 风险，代码清理未完成

关联脉络

- PR #38175 [RFC]: Support ViT Full CUDA Graph (Tracker): 本 PR 是 RFC 追踪中的具体实现之一，用于支持 Llama4 模型。

- PR #41234 [Interface] Merge encoder cudagraph interfaces: Review 中提到该 PR 将多个接口合并为单一方法，本 PR 需要同步更新以保持兼容。