PR #43082 完整报告

vllm-project/vllm

[CI] Fix "test_vit_cudagraph_[image|video][step3_vl]" failure

合并时间：2026-05-21 12:22

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43082

执行摘要

- 一句话：修复 Step3-VL 编码器 CUDA Graph 测试 OOM
- 推荐动作：值得合并，该 PR 精准修复了 CI OOM 问题，且保留了关键测试覆盖。重构的 compilation_config_overrides 模式可作为其他编码器 CUDA graph 测试的参考。CI 配置改用标签机制是良好实践。

功能与动机

PR #42224 引入的 Step3-VL 编码器 CUDA Graph 测试在 daily CI 的 L4 (22 GiB) 上因模型权重 19+ GiB 及默认编码器 cudagraph budget range 导致 OOM。PR body 直接引用 Buildkite 构建 #66759 的失败日志，目标是修复 CI 稳定性，同时保留测试覆盖。

实现拆解

1. 修改 Step3-VL 测试配置 (tests/models/multimodal/generation/test_vit_cudagraph.py)：
   - 在 VitCudagraphTestConfig 类中新增 compilation_config_overrides 字段，允许单个测试覆写编译配置。
   - 将 step3_vl 配置项中的 modalities 明确设为 ["image"]，不再支持 video 模态，使得 test_vit_cudagraph_video 执行 pytest.skip，避免实际运行。
   - 添加 compilation_config_overrides 字典，设置 encoder_cudagraph_token_budgets=[1152]，单一 bucket 恰好覆盖最大测试图片的输出 token 数（1141），避免默认自动推测生成多个 power-of-2 bucket 导致内存膨胀。
   - 通过 vllm_runner_kwargs 使用 load_format="dummy" 和 partial(dummy_hf_overrides, model_arch="StepVLForConditionalGeneration") 加载随机权重（仅保留 1 个文本层和 1 个视觉层），跳过 20 GiB 权重下载，内存占用从 19.17 GiB 降至 6.59 GiB。
2. 重构 get_compilation_config 函数：
   - 修改函数签名，接受 config: VitCudagraphTestConfig 参数，返回基础编译配置合并 config.compilation_config_overrides，实现了配置的灵活覆写。
   - 更新 test_vit_cudagraph_image 和 test_vit_cudagraph_video 中的调用，传入 config 对象。
3. CI 配置变更（.buildkite/test_areas/models_multimodal.yaml）：

- 移除在 Extended 1 中单独添加 -k step3_vl 的命令，改为在 Extended 1 中增加 --ignore models/multimodal/generation/test_vit_cudagraph.py 参数，避免在内存较小的 L4 上运行该测试。
- 新增一个独立的测试命令（在内存更大的测试环境中），通过标签 not core_model 排除 core model 测试，使用 models/multimodal/generation/test_vit_cudagraph.py -m 'not core_model' 仅运行 step3_vl 等非核心模型测试。

关键文件：

- tests/models/multimodal/generation/test_vit_cudagraph.py（模块 测试；类别 test；类型 test-coverage；符号 get_compilation_config, VitCudagraphTestConfig）：核心变更文件：通过 dummy 权重、单一 budget bucket、模态限制解决了 OOM 问题；新增 compilation_config_overrides 字段和重构 get_compilation_config 函数提供了可扩展的配置覆写模式。
- .buildkite/test_areas/models_multimodal.yaml（模块 CI；类别 infra；类型 configuration）：CI 配置变更：修改 Extended 1 测试命令以忽略该测试文件，并在独立命令中通过标签运行非 core model 测试，提升了 CI 可维护性。

关键符号：get_compilation_config, test_vit_cudagraph_image, test_vit_cudagraph_video

关键源码片段

tests/models/multimodal/generation/test_vit_cudagraph.py

核心变更文件：通过 dummy 权重、单一 budget bucket、模态限制解决了 OOM 问题；新增 compilation_config_overrides 字段和重构 get_compilation_config 函数提供了可扩展的配置覆写模式。

```
# tests/models/multimodal/generation/test_vit_cudagraph.py
# 关键配置变更

from functools import partial # 新增导入
from ....utils import dummy_hf_overrides # 新增导入

@dataclass
class VitCudagraphTestConfig:
    # ... 原有字段 ...
    compilation_config_overrides: dict = field(default_factory=dict) #
    新增：允许单个测试覆写编译配置

# Step3-VL 配置重写
MODEL_CONFIGS = {
    # ... 其他模型 ...
    "step3_vl": VitCudagraphTestConfig(
        model="stepfun-ai/Step3-VL-10B",
        modalities=["image"], # 限制仅 image 模态，video 测试将跳过
        image_prompt=step3_vl_chat_template("What is in this image?"),
        # 使用单 bucket [1152] 覆盖最大图片输出 token 数，避免多 power-of-2 bucket 导致内存膨胀
        compilation_config_overrides={
```

```
            "encoder_cudagraph_token_budgets": [1152],
        },
        # 使用 dummy 权重（仅 1 text + 1 vision layer）和随机权重，避免下载 20 GiB
        vllm_runner_kwargs={
            "load_format": "dummy",
            "hf_overrides": partial(
                dummy_hf_overrides,
                model_arch="StepVLForConditionalGeneration",
            ),
        },
    ),
}

# 重构 get_compilation_config 支持覆写
def get_compilation_config(config: VitCudagraphTestConfig):
    return {
        "cudagraph_mm_encoder": True,
        "encoder_cudagraph_max_vision_items_per_batch": 1,
        "encoder_cudagraph_max_frames_per_batch": 16,
        **config.compilation_config_overrides, # 合并覆写配置
    }
```

评论区精华

1. CI 扩展性和标签系统：
   - @DarkLight1337 质疑为何不将 step3_vl 测试放在 Extended 1（更广泛的测试 machine）中运行。@haosdent 解释 Extended 1 使用 L4 (22 GiB) GPU，直接跑会 OOM。
   - @DarkLight1337 建议改用标签（tag）而非手动匹配文件名来组织测试，提高可扩展性。@haosdent 接受建议，最终方案：在 Extended 1 中添加 --ignore 排除该文件，同时在另一条命令中使用 -m 'not core_model' 来运行。
2. 虚拟模型方案 vs. 跳过测试：
   - @Isotr0py 指出完全跳过 L4 测试存在风险，因为 step3_vl 的编码器 CUDA graph 实现将失去测试覆盖。他建议使用 hf_overrides 将模型缩小为 dummy 模型，并降低 budget size，这与最终采用的方案一致。@haosdent 随后实现了该方案。

- CI 配置组织方式：手动路径 vs 标签 (design)：改用标签机制：Extended 1 添加 --ignore 排除该文件，独立命令通过标签运行，提高可维护性。
- 是否应直接跳过 L4 上的测试 (testing)：不跳过，而是使用 dummy 权重和单一 budget bucket 使测试能在 L4 上运行。

风险与影响

- 风险：
  1. 测试覆盖退化风险：使用 dummy 权重（仅 1 层）代替完整 10B 模型，可能无法暴露与完整模型特定层数或权重分布相关的编码器 CUDA graph 问题。但测试本身仅验证 capture/replay 功能性，非精度验证，风险可控。

2. CI 配置维护负担：新的 CI 配置在 Extended 1 中添加了 --ignore 并在另一命令中运行该测试，增加了配置复杂度；但已通过标签机制和注释提高可维护性。
3. 单 bucket budget 的有效性：硬编码 token budget 为 [1152] 依赖当前最大图片 cherry_blossom 输出 token 数（1141），若未来图片解析或模型输出变化，可能需要调整。但测试灵活性已通过 compilation_config_overrides 保留。

- 影响：
  1. CI 稳定性修复：消除了 daily CI 中 step3_vl 测试在 L4 GPU 上的 OOM 失败。
  2. 测试覆盖保持：通过 dummy 模型和内存优化，保留了 step3_vl 的编码器 CUDA graph 功能测试，未完全跳过。
  3. 代码库可维护性：新增 compilation_config_overrides 字段和重构 get_compilation_config 函数，为未来其他模型测试覆写编译配置提供了通用模式。
  4. CI 配置组织：通过标签而非手动路径选择测试，提高了扩展性和可读性。 - 风险标记：测试覆盖可能退化（dummy 模型），CI 配置复杂度增加

关联脉络

- PR #42224 [MM][CG] Enable encoder Cudagraph for Step3VL: 本 PR 修复了 #42224 引入的测试在 CI L4 GPU 上的 OOM 问题。