PR #52256 完整报告

vllm-project/vllm

[ROCm][CI] Enable ViT CUDA graph tests on AMD gfx950 GPUs

合并时间：2026-08-17 09:52

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52256

执行摘要

- 一句话：在 AMD gfx950 上启用 ViT/encoder CUDA graph 测试
- 推荐动作：该 PR 值得快速浏览，主要价值在于 CI 测试扩展的组织方式：通过 is_cuda_alike() 统一平台判定、利用 --ignore 避免重复执行、将小测试整合进既有步骤而非新增独立步骤，这对维护多硬件 CI 矩阵有借鉴意义。若关心 ROCm 平台支持或 CI 效率，建议精读 .buildkite/test-amd.yaml 的改动与 review 讨论；若只关注推理内核逻辑，则无需深入。

功能与动机

PR body 明确目的：Enable ViT encoder CUDA graph (encoder CG) test coverage on AMD MI350X/MI355X (gfx950) and document the tested hardware in the design doc. 此前这些测试仅针对 CUDA 平台，通过在 ROCm 上执行 encoder CG 捕获 / 回放等测试，可以提前发现 AMD 平台上的实现差异，并为用户提供经过验证的硬件支持矩阵。

实现拆解

实现过程分四步：

1. 放宽测试平台判定：在 tests/v1/cudagraph/test_encoder_cudagraph.py 的 TestEncoderCudaGraphCaptureReplay、TestEncoderCudaGraphVideoReplay 类和 tests/models/multimodal/generation/test_vit_cudagraph.py 的 test_vit_cudagraph_image、test_vit_cudagraph_video 函数上，将 @pytest.mark.skipif(not current_platform.is_cuda(), ...) 改为 @pytest.mark.skipif(not current_platform.is_cuda_alike(), ...)，reason 同步改为 Skip if not cuda or rocm。原因是 is_cuda_alike() 同时涵盖 CUDA 与 ROCm，让 encoder/ViT CUDA graph 的捕获、回放、fallback、计数与分块等 GPU 测试可以在 AMD 平台运行。
2. 整合 CI 测试步骤：在 .buildkite/test-amd.yaml 的 MI355（gfx950）部分，将 pytest -v -s v1/cudagraph/test_encoder_cudagraph.py 并入现有的 V1 Core + KV + Metrics 步骤，并在该步骤的 source_file_dependencies 中补充 tests/v1/cudagraph；同时在 MI300 与 MI355 的 Multi-Modal Models (Standard) 4 步骤的 models/multimodal -m core_model sweep 中通过 --ignore models/multimodal/generation/test_vit_cudagraph.py 排除该文件，再在 MI355 步骤单独运行 pytest -v -s models/multimodal/generation/test_vit_cudagraph.py -m core_model，避免同一批测试被重复执行，也不新增独立 CI 步骤（回应 review 中关于减少服务器启动的诉求）。

3. 更新设计文档：在 docs/design/cuda_graphs_multimodal.md 中新增 Compatibility Matrix 小节，包含 Model x Feature 与 Model x Hardware 两张表，其中硬件表新增 AMD MI350X / MI355X 列并标为 ❓；同时注明 encoder CG 已在 MI350X 上以 ROCm 默认的 --mm-encoder-attn-backend=FLASH_ATTN 验证，并将章节 About the Performance 更名为 Benchmark Results。
4. 验证：在 MI350X/MI355X 上执行两个测试文件，PR body 声明全部通过，且 CI 配置将新加入的测试步骤设为 optional: true，避免阻塞主干。

关键文件：

- tests/models/multimodal/generation/test_vit_cudagraph.py（模块 ViT 测试；类别 test；类型 test-coverage；符号 test_vit_cudagraph_image, test_vit_cudagraph_video）：ViT encoder CUDA graph 集成测试的平台判定从 is_cuda() 放宽到 is_cuda_alike()，是本次在 ROCm 上启用测试的核心代码点。
- tests/v1/cudagraph/test_encoder_cudagraph.py（模块 编码器图；类别 test；类型 test-coverage；符号 TestEncoderCudaGraphCaptureReplay, TestEncoderCudaGraphVideoReplay）：encoder CUDA graph 核心测试类的 skip 条件同样放宽，确保捕获 / 回放 /fallback 逻辑在 gfx950 上被覆盖。
- .buildkite/test-amd.yaml（模块 CI 配置；类别 config；类型 configuration）：CI 配置把新测试整合进 MI355 现有步骤，并避免与多模态 sweep 重复，体现讨论结论。
- docs/design/cuda_graphs_multimodal.md（模块 设计文档；类别 docs；类型 documentation）：新增硬件兼容性矩阵并记录 gfx950 验证结果，支撑 ROCm 上的特性声明。

关键符号：test_vit_cudagraph_image, test_vit_cudagraph_video, TestEncoderCudaGraphCaptureReplay, TestEncoderCudaGraphVideoReplay

关键源码片段

tests/models/multimodal/generation/test_vit_cudagraph.py

ViT encoder CUDA graph 集成测试的平台判定从 is_cuda() 放宽到 is_cuda_alike()，是本次在 ROCm 上启用测试的核心代码点。

```
# tests/models/multimodal/generation/test_vit_cudagraph.py
# 变更点：将平台判定从 is_cuda() 放宽为 is_cuda_alike()，使 ViT encoder CUDA graph
测试同时覆盖 CUDA 与 ROCm。

@pytest.mark.parametrize("model_id", params_with_marks(MODEL_CONFIGS))
@pytest.mark.skipif(
    not current_platform.is_cuda_alike(), reason="Skip if not cuda or rocm"
)
def test_vit_cudagraph_image(model_id, vllm_runner, image_assets):
    config = MODEL_CONFIGS[model_id]

    if config.skip:
        pytest.skip(f"{model_id} is marked to be skipped.")
```

```
    if "image" not in config.modalities:
        pytest.skip(f"{model_id} does not support the image modality.")

    image_prompts = IMAGE_ASSETS.prompts(
        {
            "stop_sign": config.image_prompt, # type: ignore[typeddict-item]
            "cherry_blossom": config.image_prompt, # type: ignore[typeddict-item]
        }
    )
    images = [[asset.pil_image] for asset in image_assets]

    with vllm_runner(
        config.model,
        dtype=config.dtype,
        max_model_len=config.max_model_len,
        max_num_seqs=config.max_num_seqs,
        limit_mm_per_prompt={"image": 1},
        compilation_config=get_compilation_config(config),
        **config.vllm_runner_kwargs,
    ) as vllm_model:
        outputs = vllm_model.generate_greedy(
            image_prompts, config.max_tokens, images=images
        )

        # 基本校验：确保获取到响应且输出非空。
        assert len(outputs) == 2
        output_ids, output_text = outputs[0]
        assert len(output_ids) > 0
        assert len(output_text) > 0
        assert isinstance(output_text, str)
```

tests/v1/cudagraph/test_encoder_cudagraph.py

encoder CUDA graph 核心测试类的 skip 条件同样放宽，确保捕获 / 回放 /fallback 逻辑在 gfx950 上被覆盖。

```
# tests/v1/cudagraph/test_encoder_cudagraph.py
# 变更点：GPU 测试类 skip 条件改为 is_cuda_alike()，让 encoder CUDA graph 捕获 / 回放逻辑在 AMD gfx950 上接受验证。

@pytest.mark.skipif(
    not current_platform.is_cuda_alike(), reason="Skip if not cuda or rocm"
)
class TestEncoderCudaGraphCaptureReplay:
    def setup_method(self):
        self.device = torch.device("cuda:0")
        self.dtype = torch.float16
        self.model = SimpleMockViTModel().to(self.device).half()
        self.mgr = _make_manager_for_gpu(
            self.model, _BUDGETS, _MAX_BATCH, self.device, self.dtype
```

```
        )
        self.graph_pool = current_platform.graph_pool_handle()
        self.mgr.capture(graph_pool=self.graph_pool)

    # --- capture ---

    def test_capture_creates_one_graph_per_budget(self):
        assert len(self.mgr.budget_graphs["default"]) == len(_BUDGETS)
        assert set(self.mgr.budget_graphs["default"].keys()) == set(_BUDGETS)

    def test_capture_uses_supplied_graph_pool(self):
        assert self.mgr.graph_pool is self.graph_pool

    def test_clear_releases_graphs_and_pool(self):
        self.mgr.clear()
        assert self.mgr.budget_graphs == {"default": {}}
        assert self.mgr.graph_pool is None
```

评论区精华

review 的核心分歧是 CI 步骤的组织方式。AndreasKaratzas 在 .buildkite/test-amd.yaml 上两次提出 Do we need this? Can this be covered in the test group that upstream CI covers it too?，随后明确 I'm saying that I would prefer that the two test groups are integrated to the existing ones instead of adding new ones.，理由是避免为小测试启动额外服务器。shen-shanshan 最初回应可以通过 --ignore 避免重复并将新步骤设为 optional: true，在理解维护者意图后确认 Updated.，最终将 Encoder Cudagraph 测试并入 V1 Core + KV + Metrics、将 ViT CUDA Graph 测试并入既有多模态步骤。结论：不新增独立 CI 步骤，以整合方式降低 CI 启动成本。

- 是否应该新增独立 CI 步骤 (design): 最终未新增独立 CI 步骤，而是将 encoder cudagraph 测试并入 V1 Core + KV + Metrics，将 ViT cudagraph 测试并入既有多模态步骤。
- 避免测试重复执行 (design): 通过在多模态 sweep 命令中显式 --ignore test_vit_cudagraph.py，并在 MI355 步骤中安排单独运行命令，既保留覆盖又避免重复。

风险与影响

- 风险：本 PR 仅涉及测试与 CI 配置，无生产代码改动，回归风险低。但存在以下风险点：
  - 平台判定放宽到 is_cuda_alike() 后，测试同样会在其他类 CUDA 平台（如 XPU）上尝试运行，若这些平台尚未支持 encoder CUDA graph，可能出现误报或失败。不过目前仅 gfx950 目标明确。
  - 新启用的 ROCm 测试可能暴露 AMD 平台上 encoder/ViT CUDA graph 的潜在缺陷，导致 MI355 相关 CI 步骤（如 V1 Core + KV + Metrics）失败；由于步骤为 optional: true，不会阻塞主线，但可能降低覆盖信号的稳定性。
  - 多模态 sweep 中新增 --ignore 与单独运行命令的排列依赖具体 pytest 行为，若未来上游 NVIDIA CI 配置（test_areas/models_multimodal.yaml）发生变化，需要同步维护

AMD 侧配置以免重复或遗漏。

- 影响：影响范围集中在 ROCm 平台开发与 CI 流程：
  - 对用户：无功能行为变化，但文档明确了 MI350X/MI355X 上 *encoder* CUDA graph 的可用性，AMD 用户可参考兼容性矩阵选择合适的 --mm-encoder-attn-backend。
  - 对系统：CI 在 MI355 上新增了 *encoder* cudagraph 与 ViT cudagraph 测试的执行，增加约数分钟测试时间，但被整合进现有步骤且可选的，资源开销可控。
  - 对团队：ROCm 平台回归覆盖增强，可更早发现 *encoder* CUDA graph 在 gfx950 上的兼容性问题，降低维护成本。
  - 风险标记：平台判定放宽覆盖类 CUDA 后端，新启用 ROCm 测试可能失败，CI 配置需与上游 NVIDIA 同步维护

关联脉络

- PR #52441 [Bugfix][Multimodal] Keep Gemma 4 video frame counts on CPU: 同涉及 tests/models/multimodal/generation/test_vit_cudagraph.py，前者修复 Gemma 4 视频帧计数设备问题，本 PR 扩展该测试到 ROCm，属同一测试文件的持续演进。