

PR #53818 完整报告

vllm-project/vllm

[Bugfix][ROCm] Capture CUDA graphs on the current stream

合并时间: 2026-08-26 23:16

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53818>

执行摘要

- 一句话: 统一三处 CUDA 图捕获流为当前流, 修复 ROCm 下捕获失败
- 推荐动作: 该 PR 值得精读, 它揭示了 CUDA 图捕获中流上下文的重要性和潜在陷阱, 对理解 vLLM 的 cudagraph 机制及与第三方后端的交互有参考价值。建议阅读重点: `current_stream()` 的设计与 `graph_capture` 上下文管理器的配合关系。

功能与动机

在 ROCm gfx950 (MI355X) 上以 TP8 运行 Kimi-K3 (V2 model runner) 时, FULL cudagraph 捕获会因 aiter 的 split-K 工作区未初始化而失败, 导致引擎无法启动。根因是 `torch.cuda.graph()` 未传 `stream=` 时会在内部懒创建自己的捕获流, 而 warmup 和捕获在不同流上进行, 破坏了按流保存状态的 aiter 后端。PR 描述中引用了 aiter 的文档契约: 需要在捕获流上进行 eager warmup。

实现拆解

1. 定位变更入口: 从 `vllm/compilation/cuda_graph.py` 已采用 `stream=current_stream()` 的先例出发, 对齐三处遗漏的调用点。
2. 修改捕获调用: 在三处 `torch.cuda.graph()` 调用中显式传入 `stream=current_stream()`, 并引入 `vllm.utils.torch_utils.current_stream` 的导入。
3. 涉及文件:
 - `vllm/v1/worker/encoder_cudagraph.py`: `_capture_budget_graph` 方法的捕获调用。
 - `vllm/v1/worker/gpu/cudagraph_utils.py`: `capture` 方法中的捕获调用。
 - `vllm/v1/spec_decode/gemma4.py`: `_setup_centroids_cuda_graphs` 方法的捕获调用。
4. 测试配套: 本次改动未包含测试文件, 但 PR 提供了详尽的实机验证结果, 并说明修复不改变计算内容, 因此不影响精度。

关键文件:

- `vllm/v1/worker/gpu/cudagraph_utils.py` (模块 模型运行器; 类别 source; 类型 dependency-wiring; 符号 `capture`): 这是 MRV2 模型运行器的主要 CUDA 图捕获路径, FULL 和 PIECEWISE 模式均在此处捕获。修复此处直接解决大部分捕获失败问题。
- `vllm/v1/worker/encoder_cudagraph.py` (模块 编码器; 类别 source; 类型 dependency-wiring; 符号 `_capture_budget_graph`): 视觉编码器的预算批处理图捕获路

径，对多模态模型（如 Qwen）的编码器 cudagraph 生效。

- `vllm/v1/spec_decode/gemma4.py`（模块 推测解码；类别 source；类型 dependency-wiring；符号 `_setup_centroids_cuda_graphs`）：Gemma4 MTP 推测解码的质心 CUDA 图捕获路径，修复后确保 speculative decoding 在捕获时利用正确的流。

关键符号： `_capture_budget_graph`, `capture`, `_setup_centroids_cuda_graphs`

关键源码片段

`vllm/v1/worker/gpu/cudagraph_utils.py`

这是 MRV2 模型运行器的主要 CUDA 图捕获路径，FULL 和 PIECEWISE 模式均在此处捕获。修复此处直接解决大部分捕获失败问题。

```
# vllm/v1/worker/gpu/cudagraph_utils.py
# capture 方法中捕获 FULL 模式的 CUDA 图
with torch.cuda.graph(graph, self.pool, stream=current_stream()):
    forward_fn(CUDAGraphMode.NONE)
    # 捕获后合并 offloader 的复制流，避免未汇合流错误
    get_offloader().join_after_forward()
```

`vllm/v1/worker/encoder_cudagraph.py`

视觉编码器的预算批处理图捕获路径，对多模态模型（如 Qwen）的编码器 cudagraph 生效。

```
# vllm/v1/worker/encoder_cudagraph.py
# 捕获视觉编码器预算图时指定当前流
with (
    torch.inference_mode(),
    torch.cuda.graph(graph, pool=self.graph_pool, stream=current_stream()),
):
    output = self.model.encoder_cudagraph_forward(**values, path=path)
    output_buffer.copy_(output)
```

评论区精华

Review 讨论中，提交者 fanxingran 详细解释了根因，并请在 ROCm 和 nvidia 后端都有经验的维护者审阅。tjtanaa 审阅后直接批准，未产生争议性讨论。

- 捕获流的正确性 (correctness): 通过统一使用 `current_stream()` 修复，并经过实测验证。

风险与影响

- 风险：
 1. 回归风险： `current_stream()` 在未设置流时创建专用流，不会返回默认流，理论上安全；但对于依赖 `torch.cuda.graph()` 内部捕获流的第三方用例，此变更可能改变行为。
 2. 平台差异：该修复主要针对 ROCm 后端，但同样影响 NVIDIA 后端。目前描述中声明 CUDA 上行为不变（本来就是当前流），但需在 NVIDIA 环境回归验证。
 3. 测试覆盖不足：缺少自动化测试，依赖人工验证。建议后续补充针对捕获流的单元测试或集成测试。

- 影响:

1. 用户影响: 修复了 ROCm 上 Kimi-K3 等模型使用 FULL cudagraph 模式时的启动崩溃, 提升稳定性。
2. 系统影响: 改善与 aiter 等按流维护状态的后端的兼容性, 为后续优化打下基础。
3. 团队影响: 统一了代码风格, 明确了 CUDA 图捕获的流约定, 降低后续维护成本。 - 风险标记: 核心路径变更, 缺少测试覆盖, 平台差异风险

关联脉络

- PR #29207 use the same stream for cuda graph capture and replay for NCCL: 该 PR 是 vllm/compilation/cuda_graph.py 采用 stream=current_stream() 的先例, 本 PR 是对其行为的对齐。
- PR #51653 [ROCm] use MRV2 by default for Kimi-K3: 该 PR 使 Kimi-K3 在 ROCm 上默认使用 V2 运行器, 暴露了本 PR 修复的捕获问题。
- PR #53884 [Bugfix] Make Gemma4 MTP suppress_tokens masking CUDA-graph-safe: 与 Gemma4 MTP 的 CUDA 图捕获相关, 本 PR 也修改了 gemma4.py 的捕获逻辑。