PR #51222 完整报告

vllm-project/vllm

[Bugfix][EPD][Model Runner V2] Skip gather mm embeddings for encoder only instance

合并时间：2026-08-07 11:40

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51222

执行摘要

- 一句话：修复 EPD 编码实例因多余 gather 崩溃
- 推荐动作：值得精读。该 PR 展示了如何在 worker 侧而非调度器侧消除错误报告，以及如何通过拆分 encode/gather 半路径来避免无意义计算与崩溃；对 Model Runner V2 多模态路径的维护者和 EPD 部署使用者有参考价值。测试代码对 mock 契约的描述也值得借鉴。

功能与动机

PR body 指出：EPD encoder 实例一旦被调度器交给一个 EC connector 已持有的多模态项，就会在 gpu/model_runner.py:1392 → model_state.get_mm_embeddings(...) 内触发 RuntimeError: Encoder cache miss，整个实例随之崩溃，后续所有请求都得到连接错误。根因是调度器在 ec_connector.has_cache_item() 为 true 时正确跳过了该项的编码调度，但 V2 runner 在 is_encoder_only 早退之前就调用了 get_mm_embeddings()，其内部 gather 对本地 encoder cache 缺失的项抛异常，而 encoder-only 实例根本不需要这次 gather 的产物。V1 没有此 bug，因为其 producer 路径在 gather 之前就返回了。

实现拆解

1. 在 vllm/v1/worker/gpu/model_states/interface.py 的 ModelState 抽象类新增 execute_mm_encoder(scheduled_encoder_inputs) 方法：内部调用 encoder_runner.prepare_mm_inputs() 得到 mm_hashes 与 mm_kwargs，执行 encoder_runner.execute_mm_encoder() 后将输出按 mm_hash 写入 encoder_cache.encoder_outputs，全程不调用 gather_mm_embeddings()。
2. 在 vllm/v1/worker/gpu/model_runner.py 的 execute_model() 中，将 maybe_get_output 上下文内的 get_mm_embeddings 调用按 self.is_encoder_only 分流：encoder-only 实例只调用 execute_mm_encoder() 完成编码与发布，其余实例保持原逻辑。
3. 重构 vllm/v1/worker/gpu/model_states/default.py 的 DefaultModelState.get_mm_embeddings()：将原先内联的 prepare/execute/update 三行逻辑替换为一次 self.execute_mm_encoder() 调用，避免与接口新方法重复实现，语言模型实例行为不变。
4. 新增回归测试：tests/v1/worker/test_encoder_runner.py 的 test_execute_mm_encoder_caches_outputs_without_gathering 与 test_execute_mm_encoder_is_a_noop_without_scheduled_items；tests/v1/core/test_scheduler.py 的参数化测试 test_encoder_input_skipped_when_conn

ector_already_has_the_item（覆盖 ec_producer / ec_consumer），固定连接器已有缓存时调度器跳过重编码的优化。

5. 附带修复 docs/governance/committers.md 列表前缺空行（markdownlint MD032），解决 #51300 引入的 pre-commit 全量失败。

关键文件：

- vllm/v1/worker/gpu/model_states/interface.py（模块 模型状态；类别 source；类型 data-contract；符号 execute_mm_encoder）：定义新的 execute_mm_encoder() 接口方法，是本次修复的核心数据契约变更，提供只编码不 gather 的半路径。
- vllm/v1/worker/gpu/model_runner.py（模块 模型执行；类别 source；类型 core-logic）：execute_model() 中按 is_encoder_only 分流是本次修复的入口，避免 encoder-only 实例进入 gather 路径导致崩溃。
- vllm/v1/worker/gpu/model_states/default.py（模块 模型状态；类别 source；类型 refactor；符号 get_mm_embeddings）：将 get_mm_embeddings() 中内联的编码逻辑重构为调用新方法，避免重复实现，保证语言模型实例行为不变。
- tests/v1/worker/test_encoder_runner.py（模块 编码器测试；类别 test；类型 test-coverage；符号 test_execute_mm_encoder_caches_outputs_without_gathering, test_execute_mm_encoder_is_a_noop_without_scheduled_items）：新增 encode-only 行为回归测试，验证 execute_mm_encoder 写入缓存且不调用 gather，并覆盖无调度项时的 no-op 行为。
- tests/v1/core/test_scheduler.py（模块 调度器测试；类别 test；类型 test-coverage；符号 test_encoder_input_skipped_when_connector_already_has_the_item）：参数化回归测试固定调度器在连接器已有缓存时跳过重编码的优化，防止未来破坏该行为。
- docs/governance/committers.md（模块 文档；类别 docs；类型 documentation）：附带修复 #51300 引入的 markdownlint MD032 问题，保证 pre-commit 通过。

关键符号：execute_mm_encoder, get_mm_embeddings, execute_model, test_execute_mm_encoder_caches_outputs_without_gathering, test_execute_mm_encoder_is_a_noop_without_scheduled_items, test_encoder_input_skipped_when_connector_already_has_the_item

关键源码片段

vllm/v1/worker/gpu/model_states/interface.py

定义新的 execute_mm_encoder() 接口方法，是本次修复的核心数据契约变更，提供只编码不 gather 的半路径。

```
# vllm/v1/worker/gpu/model_states/interface.py
    @abstractmethod
    def get_mm_embeddings(
        self,
        scheduled_encoder_inputs: dict[str, list[int]],
        input_batch: InputBatch,
        req_states: RequestState,
    ) -> torch.Tensor | None:
```

```
        raise NotImplementedError

    def execute_mm_encoder(
        self, scheduled_encoder_inputs: dict[str, list[int]]
    ) -> None:
        """Run the multi-modal encoder and cache its outputs by `mm_hash`.

        The encode half of `get_mm_embeddings`, without the gather, for callers
        that run no language model.
        """
        # 与 get_mm_embeddings 不同的是，这里不调用 gather_mm_embeddings，
        # 因此即使本地 encoder_cache 缺少某些项，也不会抛 `Encoder cache miss`。
        # 编码输出按 mm_hash 写入缓存，供 EC connector 或下游 gather 使用。
        mm_hashes, mm_kwargs = self.encoder_runner.prepare_mm_inputs(
            scheduled_encoder_inputs
        )
        if mm_kwargs:
            encoder_outputs = self.encoder_runner.execute_mm_encoder(mm_kwargs)
            self.encoder_cache.encoder_outputs.update(zip(mm_hashes, encoder_outputs))
```

vllm/v1/worker/gpu/model_runner.py

execute_model() 中按 is_encoder_only 分流是本次修复的入口，避免 encoder-only 实例进入 gather 路径导致崩溃。

```
# vllm/v1/worker/gpu/model_runner.py（execute_model 内）
        else:
            scheduled_encoder_inputs = scheduler_output.scheduled_encoder_inputs
            if self.lora_config is not None:
                set_active_mm_loras(
                    model=self.model,
                    lora_manager=self.lora_manager,
                    encoder_cache=self.encoder_cache,
                    req_id_to_index=self.req_states.req_id_to_index,
                    lora_state=self.lora_state,
                    scheduled_encoder_inputs=scheduled_encoder_inputs,
                )
            with self.ec_connector.maybe_get_output(
                scheduler_output
            ) as ec_connector_output:
                if self.is_encoder_only:
                    # 编码器实例不运行语言模型，只负责编码并发布到连接器：
                    # 若走 get_mm_embeddings 的 gather 路径，会对连接器已持有、
                    # 但本地缓存缺失的项抛 `Encoder cache miss`，拖垮整个引擎。
                    self.model_state.execute_mm_encoder(scheduled_encoder_inputs)
                else:
                    inputs_embeds = self.model_state.get_mm_embeddings(
                        scheduled_encoder_inputs, input_batch, self.req_states
                    )
```

评论区精华

Review 中有两个核心讨论点：一是 Isotr0py 建议将新方法从 *encode_mm_inputs* 改名为 *execute_mm_encoder*，因为该方法内部要调用 *prepare_mm_inputs* 而非直接接收 mm_inputs，作者已采纳并重命名；二是 Copilot 指出测试中 *prepare_mm_inputs* 的 mock 返回值形状应与真实 EncoderRunner.*prepare_mm_inputs* 契约一致（返回 mm_hashes 与 (modality, item) 元组列表），作者相应调整了测试 mock，使其对接口变化更敏感。

- 新方法命名 *encode_mm_inputs* vs *execute_mm_encoder* (design): 采纳建议，重命名为 *execute_mm_encoder*。
- 测试 mock 返回值需匹配真实契约 (testing): 作者按建议修正测试 mock 形状，最终使用 [("image", MagicMock())]。

风险与影响

- 风险：
  1. 核心路径变更：*execute_model*() 是多模态执行主路径，新增 is_encoder_only 分支可能影响未来其他 *encoder-only* 使用场景，但目前该实例不运行语言模型，风险受控。
  2. 缓存一致性依赖：修复依赖调度器跳过调度的项一定已在 EC *connector* 中；若连接器与本地 *cache* 状态不一致（如 warm storage 重启），*encoder-only* 实例不再崩溃，但消费者侧仍可能遇到本地 *cache* miss。
  3. 测试覆盖局限：test_scheduler.py 通过 monkeypatch has_cache_item 模拟持有状态，未覆盖真实连接器交互，建议补充 EPD 集成测试观察。
  4. 文档改动仅补空行，无功能风险。 - 影响：影响范围集中在 EPD（disagg_epd_proxy 多 *encoder* 实例）部署：修复后单个 *encoder* 实例的崩溃不再拖垮整个引擎，重复图像请求与 warm storage 重启场景从 500/ 进程退出恢复为 200。对普通 *decoder*/ 语言模型实例无行为变化。团队侧，通过新增接口与回归测试把 " 连接器已有缓存时跳过重编码 " 的调度优化固化为契约，防止后续简单修复破坏该性能优化。 - 风险标记：核心路径变更，依赖 EC 缓存一致性，跨组件行为耦合，测试未覆盖真实连接器

关联脉络

- PR #51300 *docs(governance): refresh committers list, add TSC note, update project leads*: 本 PR 附带 *commit* 修复了 #51300 引入的 committers.md 列表前缺空行问题，属于跨 PR 的 *pre-commit* 修复联动。
- PR #50939 [Model Runner V2] Fix -1 placeholder draft token ids in rejection sam…: 同属 Model Runner V2 执行路径的 bugfix，且同涉及对 *encoder*/sampler 路径的边界处理，与本 PR 属于同一演进线。
- PR #51210 [ModelRunner V2] Minor indexing optimizations: 同改 vllm/v1/worker/gpu/model_runner.py，是 MRV2 执行路径性能与正确性演进的相邻 PR。