PR #41266 完整报告

vllm-project/vllm

[Frontend][Bugfix] Abort ASR engine requests on cancellation

合并时间：2026-05-10 14:51

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41266

执行摘要

- 一句话：取消 ASR 请求时中止引擎内部任务
- 推荐动作：值得精读。展示了如何在 async 服务中正确传递 CancelledError 并清理资源，同时利用 merge_async_iterators 合并多个生成器。对于处理类似音频分割的后端服务具有参考价值。

功能与动机

PR body 指出：非流式 ASR 取消时未显式中止内部引擎请求，引擎可能继续处理已取消的音频分块，造成资源泄漏和无效计算。需要确保取消时中止所有活跃的引擎任务。

实现拆解

1. 导入 merge_async_iterators：在 speech_to_text.py 中引入 vllm.utils.async_utils.merge_async_iterators，用于合并非流式分块生成器。
2. 构建引擎请求 ID 列表：根据分块数量生成唯一请求 ID（单分块复用外部请求 ID，多分块追加 -0、-1 等）。
3. try/except 包裹分块生成循环：将生成器创建循环放入 try 块，捕获 asyncio.CancelledError 后通过 asyncio.gather 调用 engine_client.abort 中止所有引擎请求，再重新抛出异常。
4. 使用 merge_async_iterators 收集非流式结果：非流式场景中，用 merge_async_iterators 同时推进所有分块生成器，按索引顺序组装最终响应。
5. 语言检测方法添加取消处理：在 _detect_language 的生成器循环外添加 try/except，捕获取消时调用 abort。
6. 新增测试文件：test_speech_to_text_cancellation.py 覆盖单分块、多分块取消以及语言检测取消场景，验证 abort 被正确调用。

关键文件：

- vllm/entrypoints/openai/speech_to_text/speech_to_text.py（模块 前端；类别 source；类型 core-logic；符号 _create_speech_to_text, _detect_language）：核心源码修改：添加取消处理和 merge_async_iterators，变更 93 行。
- tests/entrypoints/openai/speech_to_text/test_speech_to_text_cancellation.py（模块 测试；类别 test；类型 test-coverage；符号 _never_finishes,

_records_start_then_never_finishes, test_non_streaming_cancel_aborts_engine_requests, test_non_streaming_cancel_advances_all_chunk_generators）：新增测试文件：191 行，覆盖单分块、多分块取消及语言检测取消场景。

关键符号：_create_speech_to_text, _detect_language, _never_finishes, _records_start_then_never_finishes

关键源码片段

vllm/entrypoints/openai/speech_to_text/speech_to_text.py

核心源码修改：添加取消处理和 merge_async_iterators，变更 93 行。

```
# 在 _create_speech_to_text 中 engine_request_ids = [ request_id if len(engine_inputs) == 1 else f"{request_id}-{idx}" for idx in range(len(engine_inputs)) ] list_result_generator = [] try: for request_id_item, engine_input in zip(engine_request_ids, engine_inputs): # ... 创建 generator ... list_result_generator.append(generator) except asyncio.CancelledError: logger.info("Request%scancelled; aborting%dtranscription engine request(s).", request_id, len(engine_request_ids)) # 使用 gather 并发 abort 所有引擎请求 await asyncio.gather( self.engine_client.abort(engine_request_ids), return_exceptions=True, ) raise # 非流式场景使用 merge_async_iterators 合并生成器 result_generator = merge_async_iterators(*list_result_generator) async for idx, op in result_generator: # 按 idx 索引组装 chunk 结果 ... # 在 _detect_language 中 result_generator = self.engine_client.generate( prompt, sampling_params, request_id) try: final_output: RequestOutput async for final_output in result_generator: if final_output.finished: break except asyncio.CancelledError: # 取消时中止语言检测请求 await asyncio.gather( self.engine_client.abort(request_id), return_exceptions=True, ) raise
```

评论区精华

- NickLucche建议使用 merge_async_iterators 替代逐个推进生成器，以避免取消时仅第一个分块被处理的问题。作者采纳并更新了实现。
- gemini-code-assist指出 try 块应覆盖生成器循环开始前，否则已在循环期间发起的请求可能泄漏。作者随后调整了 try 范围。
- gemini-code-assist还建议在最后取消处理中也包含语言检测请求 ID。作者补全了 _detect_language 的取消处理。
- 使用 merge_async_iterators 替代逐生成器推进 (design): 作者采纳并实现，非流式分块收集改为 merge_async_iterators。
- try 块应覆盖生成器循环开始前 (correctness): 作者调整 try 块范围，将整个循环包裹在内。
- 取消处理中应包含语言检测请求 ID (correctness): 作者在 _detect_language 中也添加了类似的取消处理。

风险与影响

- 风险：
  - 核心路径变更：_create_speech_to_text 和 _detect_language 是 ASR 请求的核心方法，修改可能影响所有 ASR 请求（流式和非流式）。非流式分块顺序依赖 merge_async_iterators 的按索引组装，若出现顺序错乱可能返回乱序文本。
  - 兼容性：engine_client.abort 支持列表参数，但旧版可能不支持，需确认接口兼容性（PR 使用 asyncio.gather 包裹多次调用，安全但略低效）。
  - 性能：额外 abort 调用在取消时引入，正常路径无影响。
- 影响：
  - 用户：修复了非流式 ASR 取消时引擎资源泄漏的问题，取消请求后引擎不再继续处理，减少系统开销。
  - 系统：减少了无效计算和潜在的资源占用（如 GPU 内存）。
  - 团队：遵循了已存在的 merge_async_iterators 模式，代码风格统一。测试覆盖了核心场景，降低回归风险。
  - 风险标记：前端取消流程变更，核心用户请求路径，新增测试覆盖

关联脉络

- 暂无明显关联 PR