# PR #48333 完整报告

vllm-project/vllm

fix(entrypoints): stop resolve_items leaking in-flight media fetch tasks on partial failure

合并时间：2026-07-11 23:42

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48333

## 执行摘要

- 一句话：修复媒体获取任务泄漏问题
- 推荐动作：建议阅读。这是一个典型的 asyncio 并发控制模式改进，对理解异步任务泄露和 asyncio.gather 的正确使用有学习价值。

## 功能与动机

当单个消息包含多个图片 / 音频时，一个媒体项获取失败（如 URL 错误、超时、SSRF 拒绝、解码错误）会导致同批次其他仍在进行中的获取任务被遗弃，泄漏网络 socket 和 global_thread_pool 工作，且无法取消。PR body 提供了可复现的代码示例。

## 实现拆解

1. 修改核心获取逻辑(vllm/entrypoints/chat_utils.py): 将 AsyncMultiModalItemTracker.resolve_items 中每模态的 asyncio.gather 调用添加 return_exceptions=True 参数，使所有任务完成后才返回结果。
2. 异常收集与重新抛出: 在 gather 返回后遍历结果列表，若捕获到 BaseException 子类实例，则抛出第一个异常，确保异常传播正确。
3. 添加回归测试(tests/entrypoints/unit_tests/test_chat_utils.py): 新增 test_resolve_items_does_not_leak_tasks_on_partial_failure 测试，构造一个失败和两个慢成功的获取任务，通过 asyncio.all_tasks() 对比验证无任务泄漏。

关键文件：

- vllm/entrypoints/chat_utils.py（模块 请求处理；类别 source；类型 core-logic；符号 resolve_items）：核心修复文件，修改 AsyncMultiModalItemTracker.resolve_items 方法，将 asyncio.gather 改为 return_exceptions=True 并后处理异常。
- tests/entrypoints/unit_tests/test_chat_utils.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_resolve_items_does_not_leak_tasks_on_partial_failure, _fetch）：新增回归测试，验证 fix 后无任务泄漏。

关键符号：AsyncMultiModalItemTracker.resolve_items

## 关键源码片段

### vllm/entrypoints/chat_utils.py

核心修复文件，修改 AsyncMultiModalItemTracker.resolve_items 方法，将 asyncio.gather 改为 return_exceptions=True 并后处理异常。

```
# vllm/entrypoints/chat_utils.py (AsyncMultiModalItemTracker.resolve_items)

async def resolve_items(self) -> tuple[MultiModalDataDict | None, MultiModalUUIDDict | None]:
    if not self._items_by_modality:
        return None, None

    resolved_items_by_modality: dict[str, list[Any]] = {}
    for modality, items in self._items_by_modality.items():
        # return_exceptions=True 确保所有 tasks 都完成（或失败）后才返回，
        # 而不是在第一个异常时立即传播，导致仍在运行的 tasks 泄漏。
        results = await asyncio.gather(
            *(item() for item in items), return_exceptions=True
        )
        for result in results:
            if isinstance(result, BaseException):
                # 找到第一个异常并抛出，但此时所有 tasks 已完成。
                raise result
        resolved_items_by_modality[modality] = results

    mm_processor = (
        self.mm_processor if self._model_config.is_multimodal_model else None
    )
    return _resolve_items(
        resolved_items_by_modality,
        mm_processor,
        self._modality_order,
    )
```

## tests/entrypoints/unit_tests/test_chat_utils.py

新增回归测试，验证 fix 后无任务泄漏。

```
# tests/entrypoints/unit_tests/test_chat_utils.py ( 新增测试函数 )

@pytest.mark.asyncio
async def test_resolve_items_does_not_leak_tasks_on_partial_failure():
    """Regression test: one failing fetch must not leave sibling tasks running."""

    async def _fetch(should_fail: bool, delay: float):
        # 模拟失败或成功延迟的 fetch
        if should_fail:
            await asyncio.sleep(0.01)
            raise ValueError("simulated fetch failure")
        await asyncio.sleep(delay)
        return ("decoded", None)

    # 构造 tracker 包含 1 个失败和 2 个慢成功的 fetch
```

```
    tracker = AsyncMultiModalItemTracker(MagicMock())
    tracker._items_by_modality["image"] = [
        lambda: _fetch(True, 0),
        lambda: _fetch(False, 0.2),
        lambda: _fetch(False, 0.2),
    ]

    tasks_before = asyncio.all_tasks()
    with pytest.raises(ValueError, match="simulated fetch failure"):
        await tracker.resolve_items()

    # 确保没有新增遗留任务
    leaked_tasks = asyncio.all_tasks() - tasks_before
    assert not leaked_tasks, (
        f"resolve_items left {len(leaked_tasks)} task(s) running after "
        f"raising: {leaked_tasks}"
    )
```

## 评论区精华

无 review 评论。仅 CI pre-commit 检查发现一个格式问题（trailing blank line），由作者 ErenAta16 在第二个 commit 中修复。DarkLight1337 直接批准了 PR。

- 暂无高价值评论线程

## 风险与影响

- 风险：主要的权衡是：之前的实现会在第一个失败时立即报错（快速失败），而修复后需要等待同模态最慢的任务完成后才报错，可能增加 p99 延迟。但这是合理的取舍，因为避免了资源泄漏。
- 影响：影响所有使用多模态输入的请求路径。修复了在部分媒体获取失败时资源泄漏的 bug，提高系统稳定性和资源利用率。
- 风险标记：延迟权衡：等待慢任务增加 p99

## 关联脉络

- PR #43117 fix(processor): route MiMo-V2-Omni media fetch through MediaConnector: 同样涉及多模态媒体获取路径，修复了相关安全问题。