

PR #50221 完整报告

vllm-project/vllm

fix(security): enforce audio decode duration limit in NanoNemotronVL

合并时间: 2026-08-13 18:31

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50221>

执行摘要

- 一句话: NanoNemotronVL 音频解码加时长上限, 修复 OOM 风险
- 推荐动作: 值得精读, 但重点不在 diff 本身而在讨论区: 这是“环境变量统一防线 + 异常降级 + 定向测试”的安全修复范式。建议关注两点: depthfirst 对采样率伪造的剖析 (为什么时长检查不够)、TobyB1702 对安全修复必须带测试的坚持。合并后应跟踪 max_decode_bytes 的补齐情况。

功能与动机

PR body 明确指出: `_extract_audio_from_videos` 调用 `load_audio_pyav` 时未传 `max_duration_s`, 导致小体积压缩视频可解压成 GB 级 PCM, 服务端 OOM 崩溃。修复通过复用 `AudioMediaIO` 已有的 `VLLM_MAX_AUDIO_DECODE_DURATION_S` 环境变量防线来封堵该 DoS 入口, 并声明仅当提供该能力的 PR#49948 合并后才合入本 PR。

实现拆解

1. 安全缺口定位: `vllm/model_executor/models/nano_nemotron_vl.py` 的 `_extract_audio_from_videos` 在 `use_audio_in_video=True` 时逐视频从 `original_video_bytes` 解码音轨, 原代码 `load_audio_pyav(BytesIO(video_bytes))` 没有任何解码时长约束。
2. 防护接入: 新增 `from vllm import envs` 导入, 调用处改为传入 `max_duration_s=envs.VLLM_MAX_AUDIO_DECODE_DURATION_S`, 与 `AudioMediaIO` 的保护模式对齐; 超限抛出的异常被既有 `except Exception` 分支捕获, 该视频被标记为“无音轨” (`has_audio` 追加 `False`), 请求不会崩溃。
3. 测试配套: `tests/models/multimodal/test_nano_nemotron_vl.py` 新增辅助函数 `_make_mm_items_with_video_bytes` 构造最小 `MultiModalDataItems`; `test_extract_audio_from_videos_passes_max_duration` 用 `unittest.mock.patch` 断言 `max_duration_s` 被透传且等于环境变量; `test_extract_audio_from_videos_rejects_oversized_audio` 验证解码超限时返回空音频列表与 `[False]` 掩码。
4. 已知缺口: `commit message` 声称同时传入 `VLLM_MAX_AUDIO_DECODE_BYTES`, 但最终 diff 只有 `max_duration_s`——原因是该环境变量由 PR#49948 引入且当时尚未合入 main; depthfirst 机器人评论指出字节数防线缺失会留下采样率伪造的绕过面, 属于后续 TODO。

关键文件：

- `vllm/model_executor/models/nano_nemotron_vl.py`（模块 模型处理器；类别 `source`；类型 `data-contract`；符号 `_extract_audio_from_videos`）：安全修复主路径：为 `_extract_audio_from_videos` 的 `load_audio_pyav` 调用补上 `max_duration_s=envs.VLLM_MAX_AUDIO_DECODE_DURATION_S`，封堵解压炸弹 OOM；同时新增 `from vllm import envs` 导入。
- `tests/models/multimodal/test_nano_nemotron_vl.py`（模块 模型测试；类别 `test`；类型 `test-coverage`；符号 `_make_mm_items_with_video_bytes`，`test_extract_audio_from_videos_passes_max_duration`，`test_extract_audio_from_videos_rejects_oversized_audio`）：新增两个回归测试，验证时长上限透传与超限降级行为，是安全修复的必要配套。

关键符号：`_extract_audio_from_videos`，`test_extract_audio_from_videos_passes_max_duration`，`test_extract_audio_from_videos_rejects_oversized_audio`，`_make_mm_items_with_video_bytes`

关键源码片段

`vllm/model_executor/models/nano_nemotron_vl.py`

安全修复主路径：为 `_extract_audio_from_videos` 的 `load_audio_pyav` 调用补上 `max_duration_s=envs.VLLM_MAX_AUDIO_DECODE_DURATION_S`，封堵解压炸弹 OOM；同时新增 `from vllm import envs` 导入。

```
# 在 _extract_audio_from_videos 中逐视频抽取音轨的核心分支（改造后）。
# 安全约束：load_audio_pyav 必须传入 max_duration_s，否则一个体积很小的
# 压缩视频可被解压成数 GB 的 PCM 数据，直接拖垮服务端内存（OOM）。
for idx, metadata in enumerate(metadata_list):
    video_bytes = metadata.get("original_video_bytes")
    if video_bytes is None or len(video_bytes) == 0:
        # 缺失原始视频字节时无法抽取音轨，调用方需要以
        # keep_video_bytes=True 的方式加载视频。
        raise ValueError(
            "Cannot extract audio from video: original_video_bytes is "
            "missing or empty. When using use_audio_in_video=True, "
            "video must be loaded with keep_video_bytes=True (e.g. via "
            "the chat API with a model that sets use_audio_in_video)."
        )
    try:
        # 与 AudioMediaIO 的既有防线对齐：传 VLLM_MAX_AUDIO_DECODE_DURATION_S
        # 截断“解压炸弹”式超长音频；超限抛出的异常由下方 except 分支捕获，
        # 该视频被标记为“无音轨”，而不是让请求崩溃。
        # 注意：VLLM_MAX_AUDIO_DECODE_BYTES 字节数防线依赖 PR#49948，
        # 尚未合并 main，后续需补齐以防御采样率伪造。
        audio_items.append(
            load_audio_pyav(
                BytesIO(video_bytes),
                max_duration_s=envs.VLLM_MAX_AUDIO_DECODE_DURATION_S,
```

```
)
)
has_audio.append(True)
except Exception:
    logger.debug(
        "Video %d: no audio stream found, skipping audio extraction.",
        idx,
        exc_info=True,
    )
has_audio.append(False)
```

评论区精华

TobyB1702: main 与 PR#49948 都没有覆盖 load_audio_pyav 的测试，作为安全修复应补充 max duration 测试，可镜像 tests/multimodal/media/test_audio.py 的写法；同时提示 max_decode_bytes 只存在于 #49948 分支，需要跟踪合入。

jperezdealgaba: 补推了测试，并在 PR body 里注明仅当 #49948 合并后再合入本 PR；TobyB1702 在批准时再次提醒加“暂不合并”标签（评论中的 PR 编号疑似笔误写成了 #50221 自身）。

depthfirst-app[bot]: MEDIUM 级别建议补 max_decode_bytes=envs.VLLM_MAX_AUDIO_DECODE_BYTES——按 docs/usage/security.md，字节数防线可防御采样率伪造：头部虚高的采样率可绕过时长检查，而真实帧数仍造成多 GiB 分配。

DarkLight1337: 先提醒 rebase，作者前后 rebase 两次并触发多轮 CI (Buildkite #83179/#83182/#83657)，最终批准合并。

- max_decode_bytes 缺失，采样率伪造防线待补 (security): 本次未修复：VLLM_MAX_AUDIO_DECODE_BYTES 由 PR#49948 引入且当时未合入 main。作者与 TobyB1702 均认可需要跟踪，合入后补齐。
- 安全修复应补 load_audio_pyav 测试 (testing): 作者补推了两个针对 _extract_audio_from_videos 的 mock 测试（参数透传 + 超限降级），TobyB1702 随后批准。
- 合并顺序依赖 PR#49948 与 rebase/CI 流程 (other): PR#49948 合并后本 PR 通过 Buildkite CI (#83179/#83182/#83657) 并获得批准。

风险与影响

- 风险:
 1. 安全覆盖不完整：最终代码只传 max_duration_s，未传 max_decode_bytes。depthfirst-app[bot] 指出采样率伪造可绕过时长检查，仍可能造成多 GiB 分配；该参数依赖 PR#49948 引入的 VLLM_MAX_AUDIO_DECODE_BYTES，需在 main 合入后补齐。
 2. 行为兼容风险：超限音频被 except Exception 分支静默降级为“无音轨”，合法长音频用户可能无感知丢失声音，调试时需要 logger.debug 才可见。

3. 测试覆盖局限：新增测试为 mock 级，未覆盖 load_audio_pyav 真实解码路径；如果 load_audio_pyav 的 max_duration_s 语义在 main 上改变，测试不会暴露。
 4. 提交信息误导：commit message 声称同时传 VLLM_MAX_AUDIO_DECODE_BYTES，与实际 diff 不符，审阅和后续排查时需以代码为准。 - 影响：用户 / 系统：影响仅限 NanoNemotronVL 模型且开启 use_audio_in_video=True 的多模态推理路径；该路径从“可被恶意输入 OOM 崩溃”变为“超长音频被拒绝并降级”，未开启该选项的请求零影响。一致性：使 NanoNemotronVL 与 AudioMediaIO 共用同一个 VLLM_MAX_AUDIO_DECODE_DURATION_S 环境变量防线，多模态入口的防护语义开始统一。团队：为后续其他模型处理器接入音频解码防护提供了最小样板；遗留 max_decode_bytes 待办需要追踪，主要牵涉 PR#49948 的合入与本文件的一行补丁。
- 风险标记：OOM 解压炸弹修复，字节数防线缺失，超限音频静默降级，mock 级测试，依赖跨 PR 合并顺序

关联脉络

- PR #49948 Add audio decode safeguards (max_duration_s / max_decode_bytes): 本 PR 的合并前置依赖。PR body、commit message 与 review 讨论均明确本修复复用了其引入的 VLLM_MAX_AUDIO_DECODE_DURATION_S 和 VLLM_MAX_AUDIO_DECODE_BYTES 环境变量及 AudioMediaIO 保护；其合入 main 后才触发本 PR 的合并流程。