

PR #50827 完整报告

vllm-project/vllm

[Misc] Upgrade fastsafetensors version, fix metadata is null

合并时间: 2026-08-07 05:48

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50827>

执行摘要

- 一句话: 升级 fastsafetensors 至 0.3.3, 修复元数据缺失加载失败
- 推荐动作: 值得快速浏览而非精读。核心看点有两个: 一是依赖下限必须指向包含修复的确切版本 (而非仅放宽上限), 二是 vllm 要求文件 (.in) 与锁文件 (.txt)、setup.py extras 三处对齐的维护方式。对负责依赖管理和 CI 维护的工程师有参考价值。

功能与动机

PR body 描述: 使用 `--load-format fastsafetensors` 加载 MiniMaxAI/MiniMax-M3-MXFP8 时报 `SafeTensorsMetadataKeyError: 'data_offsets'`, 该问题由上游 `foundation-model-stack/fastsafetensors#89` 修复。因此需要升级 fastsafetensors 到包含修复的 0.3.3。njhill 的补充提交进一步强调: 修复只在 0.3.3 中, `>= 0.3.2` 会让解析器选中坏版本, 必须统一提升下限。

实现拆解

1. 运行时依赖入口: 修改 setup.py 中 `extras_require["fastsafetensors"]` 的下限为 `>= 0.3.3`, 并同步 requirements/cuda.txt、requirements/rocm.txt。这是用户安装 vllm 后实际解析到的依赖, 确保新装环境拿到修复版本。
2. 测试依赖源文件: requirements/test/cuda.in 与 requirements/test/rocm.in 中 `fastsafetensors>=0.3.2` 改为 `>=0.3.3`, 让 CUDA/ROCm CI 环境与生产运行行为一致, 避免测试环境复现旧版问题。
3. 锁文件同步: requirements/test/cpu.txt、requirements/test/cuda.txt、requirements/test/rocm.txt 与 requirements/test/nightly-torch.txt 中精确版本由 0.3.2 更新为 0.3.3, 保证可复现安装与 nightly 流水线一致。
4. 验证: PR body 提供 `vllm serve MiniMaxAI/MiniMax-M3-MXFP8 -tp=8 --load-format fastsafetensors` 的完整启动日志, 服务正常拉起; Buildkite CI 触发两轮 (#82458、#82589) 通过。本次未新增测试文件, 回归覆盖依赖既有 fastsafetensors 加载路径。

关键文件:

- setup.py (模块 包配置; 类别 source; 类型 core-logic): 定义 `vllm[fastsafetensors]` extra 的依赖下限, 是运行时安装入口, 本次修复的核心生效点。
- requirements/test/cuda.in (模块 测试依赖; 类别 test; 类型 test-coverage): CUDA CI 测试环境的依赖源文件, 与运行时保持相同下限, 防止测试环境解析到旧坏版本。

- requirements/test/rocm.in (模块 测试依赖; 类别 test; 类型 test-coverage) : ROCm CI 测试依赖源文件同步升级, 保证 AMD 平台的测试环境同样使用修复版本。
- requirements/cuda.txt (模块 依赖锁文件; 类别 docs; 类型 configuration) : CUDA 运行时锁文件, 记录解析后的精确版本, 必须与源文件下限同步以保证可复现安装。
- requirements/test/nightly-torch.txt (模块 依赖锁文件; 类别 docs; 类型 configuration) : nightly 流水线锁文件同步更新, 避免 nightly 环境解析到旧版本而遗漏修复。

关键符号: 未识别

关键源码片段

setup.py

定义 vllm[fastsafetensors] extra 的依赖下限, 是运行时安装入口, 本次修复的核心生效点。

```
setup(
    # 静态元数据应尽量放在 pyproject.toml, 这里保留部分运行时配置
    version=get_vllm_version(),
    ext_modules=ext_modules,
    rust_extensions=rust_extensions,
    install_requires=get_requirements(),
    extras_require={
        # 其他可选依赖省略
        "tensorizer": ["tensorizer==2.10.1"],
        # fastsafetensors 0.3.3 才包含 data_offsets 修复;
        # 低于该版本会导致 --load-format fastsafetensors 加载
        # MXFP8 模型 (如 MiniMax-M3-MXFP8) 时抛出
        # SafeTensorsMetadataKeyError: 'data_offsets'
        "fastsafetensors": ["fastsafetensors >= 0.3.3"],
        "instanttensor": ["instanttensor >= 0.1.9"],
        # ...
    },
)
```

评论区精华

njhill 的补丁说明: 提交 [89318d0b](#) 指出 `data_offsets` 修复只在 0.3.3 中, `>= 0.3.2` 仍可能装到坏版本, 因此将运行时、extra 与测试依赖全部提升, 并同步锁文件。批准意见: njhill 在 review 中表示 Thanks @lengrongfu, I pushed a commit to update the version in a few more places. 并给予 APPROVED。claude[bot] 提示: 来自 fork 的 PR 不自动 review, 需维护者手动审阅。

- fastsafetensors 下限需提升到 0.3.3 而非 0.3.2 (correctness): 统一将下限提升到 `>= 0.3.3`, 锁文件同步更新到 0.3.3, PR 获得 njhill 批准。
- fork PR 自动 review 被禁用 (other): njhill 手动审阅并批准。

风险与影响

- 风险:

1. 测试覆盖不足：本次仅调整依赖版本，没有为 data_offsets 场景新增回归测试，若将来 fastsafetensors 再次回归，CI 无法直接捕获。
2. 多平台同步风险：改动同时涉及 cuda、rocm、cpu 三套 requirements 与 nightly 锁文件，若遗漏其他 lockfile（如 docker 构建缓存中的固定版本），对应平台仍会解析到 0.3.2。
3. 安装缓存：用户环境中若已缓存 0.3.2 的 wheel 或旧锁文件，升级可能不会自动刷新，需要显式更新。
4. 兼容性：0.3.3 为修复版本，API 未变，回归风险低，但仍依赖上游 fastsafetensors 的二进制覆盖（如 ROCm、CPU 平台）。- 影响：对用户：修复 --load-format fastsafetensors 加载 MXFP8 模型（如 MiniMaxAI/MiniMax-M3-MXFP8）时的 SafeTensorsMetadataKeyError 崩溃，安装新版本 vllm 或依赖后即可解决。对系统：无运行时逻辑改动，仅依赖解析版本变化。对团队：属于低风险常规维护，锁文件三处同步（extras、.in、.txt）为后续依赖升级提供了可复制的操作模式。- 风险标记：依赖版本升级，无新增测试覆盖，多平台同步

关联脉络

- 暂无明显关联 PR