PR #43625 完整报告

vllm-project/vllm

[ROCm] Bump fastsafetensors to v0.3.2 from PyPI, remove git source build

合并时间：2026-06-04 22:30

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43625

执行摘要

- 一句话：fastsafetensors 升级至 v0.3.2，移除 ROCm git 构建
- 推荐动作：该 PR 属于基础设施清理，改动量小但意义明确，建议合并。但建议在合并前确认非 x86 平台（如 ARM）上 fastsafetensors v0.3.2 的兼容性，或保留 platform_machine 过滤。

功能与动机

此前，fastsafetensors 的 PyPI wheel 仅支持 CUDA，导致 ROCm 用户必须从 Git 构建。上游 PR（foundation-model-stack/fastsafetensors#78）实现了通用 wheel，通过运行时 dlopen 检测 CUDA 或 ROCm runtime，因此可以移除 ROCm 的源码构建方案，并统一所有平台的依赖来源。

实现拆解

1. 在 requirements/test/rocm.in 中将 fastsafetensors 由 git 源替换为 fastsafetensors>=0.3.2，并移除附注注释。
2. 在 setup.py 中将 *extras_require* 中 fastsafetensors 的下限从 >=0.2.2 提升至 >=0.3.2。
3. 在 requirements/test/cuda.in、requirements/cuda.txt、requirements/test/nightly-torch.txt 中统一版本下限为 >=0.3.2。
4. 在 requirements/rocm.txt 中新增 fastsafetensors>=0.3.2 条目（此前 ROCm 依赖不含此包，因其通过 git 构建）。
5. 由 pip-compile 重新生成 requirements/test/rocm.txt 和 requirements/test/cuda.txt 锁文件，更新哈希和版本。
6. 验证：在 ROCm MI300X 上成功加载 Qwen3-0.6B 模型全部 311 个 tensor。 注意：没有新增测试代码，但作者提供了手动测试脚本，审核者 tjtanaa 也在 MI355X 上验证通过。

关键文件：

- requirements/test/rocm.in（模块 ROCm 测试依赖；类别 test；类型 test-coverage）：核心变更文件：将从 Git 源码构建的依赖替换为 PyPI 包，是 PR 的主要动机所在。
- setup.py（模块 构建配置；类别 source；类型 core-logic）：定义了可选依赖 fastsafetensors 的版本下限，同步提升。

- requirements/rocm.txt（模块 ROCm 运行时依赖；类别 docs；类型 documentation）：新增 fastsafetensors 依赖条目，此前 ROCm 运行时依赖中不包含此包（因为它通过 git 构建安装）。
- requirements/test/cuda.in（模块 CUDA 测试依赖；类别 test；类型 test-coverage）：同步升级 CUDA 测试依赖中 fastsafetensors 版本下限，并移除了 platform_machine 过滤。
- requirements/test/rocm.txt（模块 ROCm 测试锁定；类别 docs；类型 documentation）：锁文件更新，反映 fastsafetensors 来源从 git 改为 PyPI，版本固定为 0.3.2。
- requirements/test/cuda.txt（模块 CUDA 测试锁定；类别 docs；类型 documentation）：锁文件更新，fastsafetensors 版本从 0.2.2 升至 0.3.2。
- requirements/cuda.txt（模块 CUDA 运行时锁定；类别 docs；类型 documentation）：CUDA 运行时依赖锁文件，fastsafetensors 版本下限提升。
- requirements/test/nightly-torch.txt（模块 夜间测试锁定；类别 docs；类型 documentation）：夜间 torch 测试依赖锁文件，同步版本下限。

关键符号：未识别

关键源码片段

setup.py

定义了可选依赖 fastsafetensors 的版本下限，同步提升。

```
# setup.py (extras_require 片段 )
extras_require={
    ...
    "tensorizer": ["tensorizer==2.10.1"],
    "fastsafetensors": ["fastsafetensors >= 0.3.2"], # 从 0.2.2 提升，因 v0.3.2 提供通用 wheel
    "instanttensor": ["instanttensor >= 0.1.5"],
    ...
}
```

评论区精华

AndreasKaratzas 在 requirements/test/rocm.in 的评论中询问：“我们现在是否提供 ROCm wheels？因为之前这是个问题。” tjtanaa 回复：“PyPI wheel 现在与硬件无关，它在运行时加载 NVIDIA/hip runtime 库。我们不需要为 FST 构建自定义 ROCm wheel 了。我自己在 MI355X 上也验证了可工作。” 这确认了迁移的核心前提，消除了审核者的疑虑。

- ROCm 版本的 fastsafetensors 是否已支持 PyPI wheel？ (question): PyPI v0.3.2 已支持 ROCm，可以安全切换。

风险与影响

- 风险：主要风险在于 fastsafetensors v0.3.2 的通用 wheel 在某些 ROCm 版本上可能存在意外的运行时加载失败。但 tjtanaa 已在 MI300X 和 MI355X 上验证了模型加载成功，降低了此风险。此外，PR 将 CUDA 一侧的 platform_machine 约束也移除了（原为 x86_64），可能影响 ARM64 等非 x86 平台，需要确认 fastsafetensors 在这些平台上的兼容性。

另一个风险是该变更依赖上游正确实现了运行时检测，如果出现回归则可能影响所有平台。

- 影响：对 ROCm 用户：安装 vLLM 时不再需要从 Git 构建 fastsafetensors，安装流程简化，构建时间缩短。对 CUDA 用户：无功能影响，仅依赖版本从 0.2.2 升至 0.3.2，应向前兼容。对 vLLM 项目：统一了 CUDA 和 ROCm 的依赖管理，减少了维护分支的成本。
- 风险标记：上游依赖变更，非 x86 平台兼容

关联脉络

- 暂无明显关联 PR