

PR #43530 完整报告

vllm-project/vllm

Fix CuPy runtime deps and restore humming

合并时间: 2026-05-26 20:59

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/43530>

执行摘要

- 一句话: 修复 CuPy 14.1.0 的 pytest 冲突并恢复 humming 支持
- 推荐动作: 建议在 humming.py 中保留适当的安装提示 (例如在类 `__init__` 中重新添加检查), 或迁移至更优雅的延迟错误报告。同时跟踪 cupy 14.1.1 或更高版本, 尽快将版本限制从 `<14.1.0` 放宽为 `!=14.1.0`。

功能与动机

PR body 指出 #43492 临时回退了 #42540, 但 #43528 显示在移除 humming 后相同的 pytest 失败仍然存在。根因是 vllm-openai:nightly 默认安装 KV connectors, lmcache 拉取了最近发布的 cupy-cuda13x==14.1.0, 该版本在 PyTorch 自定义 op 注册时通过 inspect 遍历已加载模块并导入 pytest。

实现拆解

1. 限制 CuPy 版本: 在 requirements/kv_connectors.txt 中添加 `cupy-cuda13x < 14.1.0`, 避免导入 pytest 的 cupy 14.1.0。
2. 恢复 humming 依赖: 在 requirements/cuda.txt 中添加 `humming-kernels[cu13]==0.1.2`。
3. CUDA 12 变体重写: 在 setup.py 的 CUDA 12 路径中添加 humming-kernels[cu13] 到 [cu12] 的重写逻辑。
4. Docker 同步: 在 docker/Dockerfile 的两处 sed 命令中添加 humming-kernels 重写, 并修正 nvidia-cutlass-dsl 重写模式 (从仅匹配 `>=` 改为通用匹配)。
5. 重构 humming 导入: 将 vllm/model_executor/layers/quantization/humming.py 中的 `try/except ModuleNotFoundError` 替换为 `if current_platform.is_cuda()` 条件导入, 移除 `assert_humming_available()` 函数及其在 HummingConfig.__init__ 中的调用。

关键文件:

- vllm/model_executor/layers/quantization/humming.py (模块 量化模块; 类别 source; 类型 core-logic; 符号 `assert_humming_available`, `HummingConfig.init`): 核心量化模块; 导入方式重构 (条件导入替代 try/except), 移除 `assert_humming_available` 函数及调用
- setup.py (模块 构建脚本; 类别 source; 类型 core-logic): 构建脚本; 添加 humming-kernels 的 CUDA 12 变体重写

- docker/Dockerfile (模块 Docker; 类别 infra; 类型 infrastructure) : Docker 构建; 同步 humming-kernels 重写并修复 cutlass-dsl 重写模式
- requirements/cuda.txt (模块 依赖配置; 类别 config; 类型 configuration) : CUDA 依赖声明; 添加 humming-kernels 依赖项
- requirements/kv_connectors.txt (模块 依赖配置; 类别 config; 类型 configuration) : KV connectors 依赖声明; 添加 CuPy 版本限制以规避 pytest 冲突

关键符号: `assert_humming_available`, `HummingConfig.init`

关键源码片段

vllm/model_executor/layers/quantization/humming.py

核心量化模块; 导入方式重构 (条件导入替代 try/except), 移除 `assert_humming_available` 函数及调用

```
from vllm.platforms import current_platform

# 仅在 CUDA 平台上尝试导入 humming 模块
# 若不在 CUDA 平台或不满足依赖, HummingMethod 将在后续调用中引发 NameError
if current_platform.is_cuda():
    from humming.dtypes import DataType
    from humming.layer import HummingMethod
    from humming.schema import (
        BaseInputSchema,
        BaseWeightSchema,
        HummingInputSchema,
        HummingWeightSchema,
    )
    from humming.utils.weight import quantize_weight
    from vllm.model_executor.layers.fused_moe.experts.fused_humming_moe import (
        BatchedHummingGroupedExperts,
        HummingGroupedExperts,
        HummingIndexedExperts,
        get_humming_moe_gemm_type,
    )

    if TYPE_CHECKING:
        from humming.schema import (
            BaseInputSchema,
            BaseWeightSchema,
            HummingInputSchema,
            HummingWeightSchema,
        )
        from vllm.model_executor.models.utils import WeightsMapper

# 类定义中 __init__ 不再调用 assert_humming_available()
class HummingConfig(QuantizationConfig):
    packed_modules_mapping: dict[str, list[str]] = {}
```

```
def __init__(self, full_config: dict[str, Any] | None = None):
    # 之前此处调用了 assert_humming_available(), 现在删除了该检查
    # 如果 humming 未安装且平台为 CUDA, 后续访问 HummingMethod 会抛出 NameError
    self.full_config: dict[str, Any] = full_config or {}
```

评论区精华

- 错误处理退化 (gemini-code-assist) : 移除 `assert_humming_available` 后, 未安装 humming 的用户会遭遇 `NameError` 而非清晰提示。该评论未得到作者回应, PR 最终合并。
- CuPy 限制范围 (mgoin vs mmangkad) : mgoin 提议将 `cupy` 限制放到基础 deps, 但作者确认 humming 不直接依赖 `cupy`, 仅因 `lmcache` 引入, 限定在 `kv_connectors.txt` 足够。结论: 维持原方案。
- CuPy 上游修复 (leofang) : CuPy 14.1.1 已修复 `pytest` 导入问题, 建议将限制改为 `!<14.1.0`。PR 合并时未更新, 留待后续处理。
- 错误处理回归: `assert_humming_available` 被移除 (correctness): 作者未回应, PR 合并时未恢复。虽然条件导入避免了 `ModuleNotFoundError`, 但错误信息仍不够友好。
- CuPy 版本限制应否放入基础依赖 (design): 维持限制在 `kv_connectors.txt`, 作者理由被接受。
- CuPy 14.1.1 上游修复通知 (other): PR 合并时未更新, 留待后续 PR 处理。

风险与影响

- 风险:
 - 错误提示退化: `HummingConfig.__init__` 中移除了 `assert_humming_available()`, 若用户在不支持或未安装 humming 的环境下尝试量化, 将得到模糊的 `NameError`, 而非可操作的安装提示。
 - 依赖版本锁定: `cupy-cuda13x < 14.1.0` 会阻止未来可能兼容的更新, 需要手动跟进上游修复并放宽限制。
 - Docker 与 `setup.py` 逻辑重复: `Dockerfile` 中的 `sed` 重写与 `setup.py` 的逻辑重复, 且 `sed` 基于行精确匹配, 若 `requirements/cuda.txt` 格式变化 (如换行、注释变更), 可能导致重写失效。
 - 平台兼容性: 条件导入 `if current_platform.is_cuda()` 使非 CUDA 平台无法使用 humming, 但 humming 本身可能在其他平台不可用, 该变化合理但需确保不产生副作用。
- 影响:
 - 用户影响: 使用 humming 量化的用户恢复功能, 使用 KV connectors 的用户避免 `pytest` 冲突。未安装 humming 的用户若错误触发相关代码, 错误信息变差。
 - 系统影响: CUDA 12 Docker 镜像现在正确安装 cu12 变体, 与 `pip` 安装行为一致。
 - 团队影响: 依赖管理复杂度增加, 需跟踪 `cupy` 上游修复并放宽限制。
 - 风险标记: 错误检查退化, 依赖版本锁定, Docker 与 `setup.py` 逻辑重复

关联脉络

- PR #42540 Add humming-kernels support: 本 PR 恢复的就是 #42540 引入的 humming 量化功能
- PR #43480 Humming import issue investigation: 该 PR 触发对 humming 导入问题的调查
- PR #43492 Temporarily revert humming: 因 #43480 而临时回退 humming, 本 PR 确认根因并非 humming 故恢复
- PR #43528 Pytest failure after humming removal: 显示即使移除 humming 后相同的 pytest 失败仍然存在, 证实根因在 CuPy