PR #46567 完整报告

vllm-project/vllm

Fix model info cache for package models

合并时间：2026-06-29 17:17

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46567

执行摘要

- 一句话：修复包式模型注册的缓存失效问题
- 推荐动作：该 PR 值得精读，特别是 _get_modelinfo_module_hash 的设计决策。对于其他需要缓存包模块哈希的场景，此模式可复用。

功能与动机

修复模型注册缓存键仅覆盖 __init__.py，导致包式注册模型（如 MiniMax-M3 PP 工作）的子模块变更时，缓存的 capability 元数据不会失效。这一问题在审查 #45810 时发现。

实现拆解

1. 新增 _get_modelinfo_module_hash 静态方法：在 vllm/model_executor/models/registry.py 的 _LazyRegisteredModel 类中新增，接受 model_path 参数，若文件名是 __init__.py，则对包目录下所有 .py 文件（rglob("*.py")）按字典序读取内容并计算累积哈希；否则仅对单文件哈希。
2. 替换 inspect_model_cls 中的哈希逻辑：将原来直接对单文件 safe_hash 的调用改为调用 self._get_modelinfo_module_hash(model_path)，复用新逻辑。
3. 新增回归测试：在 tests/models/test_registry.py 中新增 test_lazy_modelinfo_package_hash_includes_submodules，创建一个临时包目录，修改子模块文件后断言两次哈希不同，确保缓存正确失效。
4. 导入调整与测试优化：测试文件新增 _LazyRegisteredModel 导入；最后一次提交修复了 import 顺序。

关键文件：

- vllm/model_executor/models/registry.py（模块 模型注册；类别 source；类型 data-contract；符号 _get_modelinfo_module_hash）：核心变更文件，新增 _get_modelinfo_module_hash 静态方法并替换 inspect_model_cls 中的哈希逻辑，是影响缓存正确性的关键改动。
- tests/models/test_registry.py（模块 测试；类别 test；类型 test-coverage；符号 test_lazy_modelinfo_package_hash_includes_submodules）：新增回归测试 test_lazy_modelinfo_package_hash_includes_submodules，通过临时包目录验证子模块变更会导致哈希变化，是验证正确性的关键。

关键符号：_get_modelinfo_module_hash

关键源码片段

vllm/model_executor/models/registry.py

核心变更文件，新增 _get_modelinfo_module_hash 静态方法并替换 inspect_model_cls 中的哈希逻辑，是影响缓存正确性的关键改动。

```
@staticmethod
def _get_modelinfo_module_hash(model_path: Path) -> str:
    if model_path.name == "__init__.py":
        # Package entry points often re-export classes implemented in
        # submodules, so include the package contents in the cache key.
        module_paths = sorted(model_path.parent.rglob("*.py"))
        root_path = model_path.parent
    else:
        # 单文件模型保持原有单文件哈希路径
        module_paths = [model_path]
        root_path = model_path.parent

    hasher = safe_hash(b"", usedforsecurity=False)
    for path in module_paths:
        # 使用相对路径作为标识，避免绝对路径泄露
        hasher.update(path.relative_to(root_path).as_posix().encode("utf-8"))
        hasher.update(b"\0")
        # 读取文件内容参与哈希
        hasher.update(path.read_bytes())
        hasher.update(b"\0")
    return hasher.hexdigest()
```

tests/models/test_registry.py

新增回归测试 test_lazy_modelinfo_package_hash_includes_submodules，通过临时包目录验证子模块变更会导致哈希变化，是验证正确性的关键。

```
def test_lazy_modelinfo_package_hash_includes_submodules(tmp_path):
    # 模拟一个包式模型注册：__init__.py 从 .model 子模块导入 Model
    package_dir = tmp_path / "model_package"
    package_dir.mkdir()
    init_file = package_dir / "__init__.py"
    init_file.write_text("from .model import Model\n", encoding="utf-8")
    model_file = package_dir / "model.py"
    model_file.write_text("class Model: pass\n", encoding="utf-8")

    # 计算首次哈希
    first_hash = _LazyRegisteredModel._get_modelinfo_module_hash(init_file)

    # 修改子模块文件，预期哈希变化
    model_file.write_text(
        "class Model:\n    supports_pp = True\n", encoding="utf-8"
```

```
)
second_hash = _LazyRegisteredModel._get_modelinfo_module_hash(init_file)

# 断言：子模块变更应导致缓存失效
assert first_hash != second_hash
```

评论区精华

PR 讨论较少，Hmellor 批准后仅提及 CI 失败的 Hugging Face 429 问题（不相关）。无技术争议。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更集中在缓存键计算逻辑，不影响功能正确性。包遍历 rglob("*.py") 在包目录较大时可能引入轻微性能开销，但模型信息缓存本身仅在首次加载或文件变更时计算，频率极低。测试覆盖了包场景和单文件场景的退化路径。
- 影响：影响范围小，仅涉及 _LazyRegisteredModel 的缓存哈希计算。对用户无直接感知，但能避免因缓存误命中导致模型 capability 信息过时而引发的调试困扰。团队在维护硬件特定模型（如 vllm.models.minimax_m3）时会从中受益。
- 风险标记：低风险

关联脉络

- PR #45810 [MiniMax-M3] PP support: PR 中提及该模型注册是包式的，且发现此缓存问题的场景。