PR #40636 完整报告

vllm-project/vllm

Fix test_startup.py for torch 2.12

合并时间：2026-04-23 03:31

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40636

执行摘要

- 一句话：修复 PyTorch 2.12 下编译启动测试因版本检测和缓存行为变化导致的失败。
- 推荐动作：该 PR 值得快速浏览，重点关注版本检测的调整逻辑和测试预期的条件化设计。对于维护 vLLM 与 PyTorch 版本兼容性的团队，可学习如何优雅处理开发版与正式版版本号差异。无需深入阅读源码，但可注意 is_torch_equal_or_newer 函数的使用模式。

功能与动机

PR 动机源于 PyTorch 2.12 发布版测试中出现的回归问题（关联 Issue #180912）。在 torch 2.12.0 下，vLLM 的 H100 编译启动测试失败，表现为 warm_artifacts_saved 预期为 0 但实际为 4，以及 KeyError: None 错误。PR body 指出，测试在 torch 2.12 下观察到 warm_artifacts_saved = 0 和 warm_artifacts_loaded = 4，这与之前版本的行为不同，因此需要调整测试预期以适配新版本。同时，这也可能修复了 Issue #38051 中提到的 warm start 编译时间问题。

实现拆解

1. 调整测试预期以适配 PyTorch 2.12 的缓存行为变化：在 tests/compile/h100/test_startup.py 中，修改了 deepseek_v3.2 和 kimi_k2.5 两个测试用例的 ModelStartupSpec。将 warm_artifacts_saved 和 warm_artifacts_loaded 的硬编码值改为条件表达式，当 PyTorch 版本 >= 2.12.0 时，warm_artifacts_saved 设为 0，warm_artifacts_loaded 设为 4；否则保持原值（4 和 0）。这反映了 PyTorch 2.12 中 warm 缓存不再保存新 artifact 而是加载现有缓存的行为变化。
2. 修复版本检测逻辑以匹配开发版版本号：在 vllm/env_override.py 中，将条件 not is_torch_equal_or_newer("2.12.0") 改为 not is_torch_equal_or_newer("2.12.0.dev")。这是因为从源码构建的 PyTorch 2.12 版本号可能带有 "a0" 后缀（alpha 版本），被识别为低于 "2.12.0"，导致本应在 2.12 中启用的补丁逻辑被错误跳过。
3. 同步更新其他测试中的版本检测：在 tests/compile/test_dynamic_shapes_compilation.py 中，将 get_test_models 函数中的条件 is_torch_equal_or_newer("2.12.0") 改为 is_torch_equal_or_newer("2.12.0.dev")，确保在 PyTorch 2.12 下正确添加 Qwen3-4B 模型到测试列表中。

关键文件：

- tests/compile/h100/test_startup.py（模块 启动测试；类别 test；类型 test-coverage）：核心测试文件，直接修复 PyTorch 2.12 下 warm 缓存行为变化导致的测试失败。
- vllm/env_override.py（模块 环境覆盖；类别 source；类型 core-logic）：源码文件，修复版本检测逻辑，确保 PyTorch 2.12 开发版下特定补丁正确启用。
- tests/compile/test_dynamic_shapes_compilation.py（模块 动态形状编译；类别 test；类型 test-coverage）：测试文件，同步更新版本检测，确保测试模型列表在 PyTorch 2.12 下正确包含 Qwen3-4B。

关键符号：is_torch_equal_or_newer

关键源码片段

tests/compile/h100/test_startup.py

核心测试文件，直接修复 PyTorch 2.12 下 warm 缓存行为变化导致的测试失败。

```
# 在 tests/compile/h100/test_startup.py 中，ModelStartupSpec 的修改片段
pytest.param(
    ModelStartupSpec(
        model="deepseek-ai/DeepSeek-V3.2",
        hf_overrides=_SMALL_MOE_OVERRIDES,
        cold_artifacts_saved=4,
        # https://github.com/vllm-project/vllm/issues/38051
        # 在 PyTorch 2.12 及以上版本，warm 启动时不再保存新 artifact，而是加载现有缓存
        warm_artifacts_saved=0 if is_torch_equal_or_newer("2.12.0") else 4,
        warm_artifacts_loaded=4 if is_torch_equal_or_newer("2.12.0") else 0,
    ),
    id="deepseek_v3.2",
),
# 类似修改也应用于 kimi_k2.5 测试用例
```

vllm/env_override.py

源码文件，修复版本检测逻辑，确保 PyTorch 2.12 开发版下特定补丁正确启用。

```
# 在 vllm/env_override.py 中，版本检测条件的修改
if is_torch_equal_or_newer("2.10.0") and not is_torch_equal_or_newer("2.12.0.dev"):
    # 此补丁逻辑在 PyTorch 2.10.0 至 2.12.0.dev 之前版本启用
    # 修改前使用 "2.12.0"，但从源码构建的 PyTorch 2.12 可能带 alpha 后缀（如 2.12.0a0），
    # 导致版本比较错误，故改为 "2.12.0.dev" 以正确匹配开发版
    import builtins as _builtins
    import pickle
    # ... 后续补丁代码
```

评论区精华

review 讨论较少，主要聚焦于版本检测的细节。作者 angelayi 在 tests/compile/test_dynamic_shapes_compilation.py 的评论中解释："When I built the release/2.12.0 from source it has the a0 suffix meaning alpha release, so the versioning treats it as less than 2.12. Changing this to be 2.12.0.dev seems to fix the

issue"。这澄清了将 "2.12.0" 改为 "2.12.0.dev" 的原因，即从源码构建的 PyTorch 2.12 可能带有 alpha 后缀，导致版本比较出错。其他 review 为自动化 bot 的占位评论，无实质争议。

- 版本检测从 2.12.0 改为 2.12.0.dev 的原因 (correctness): 将硬编码版本号改为 '2.12.0.dev' 以正确匹配开发版版本号。

风险与影响

- 风险：技术风险较低：
  - 回归风险：修改仅限于版本检测和测试预期，未触及核心编译或模型逻辑，但需确保条件表达式 is_torch_equal_or_newer("2.12.0") 和 is_torch_equal_or_newer("2.12.0.dev") 在所有 PyTorch 版本（包括正式版、RC 版、源码构建版）下行为一致，否则可能导致补丁误启用或测试误判。
  - 测试覆盖风险：调整测试预期后，若 PyTorch 2.12 的缓存行为后续修复或变化，测试可能无法及时捕获，需依赖持续集成监控。
  - 兼容性风险：无，变更保持向后兼容，仅适配新版本行为。
- 影响：影响范围有限：
  - 对用户：无直接影响，纯测试和内部版本检测调整。
  - 对系统：确保编译启动测试在 PyTorch 2.12 下通过，支持 vLLM 升级到 PyTorch 2.12（如 PR #40077），避免阻塞版本升级。
  - 对团队：修复了 CI 测试失败，提升开发体验；揭示了 PyTorch 版本号处理细节，为未来类似问题提供参考。
  - 风险标记：版本检测逻辑调整，测试预期条件化

关联脉络

- PR #40077 [Upgrade] torch 2.12: 关联 PR，此 PR 修复的测试失败正阻塞 PyTorch 2.12 升级。