PR #49960 完整报告

vllm-project/vllm

[CPU] Fix torch.compile crash from torch.accelerator.synchronize on CPU-only hosts

合并时间：2026-08-03 19:48

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49960

执行摘要

- 一句话：CPU-only 主机 torch.compile 崩溃修复：empty_cache 改用独立 no-op
- 推荐动作：值得精读的小型修复：改动仅 13 行，但完整展示了一个编译期崩溃 -> 最小复现 -> 根因定位（对象身份别名）-> 极简修复的调试链路。对维护 CPU worker 补丁与接触 torch.compile 图形捕获的同学有参考价值；也提示 Dynamo 对 torch 模块 API 的追踪可能依赖函数对象身份，写 monkey patch 时需避免别名共享。

功能与动机

PR body 指出：CPU-only 主机上 torch.compile（如 gpt-oss）在 aot_compile 期间崩溃，报 AssertionError: No accelerator available for torch.accelerator.synchronize；synchronize 在 CPU 上无加速器可用，eager 模式会 raise，Dynamo 的 handle_synchronize 也会断言。讨论中 ganeshr10 进一步定位到真正触发点是 vllm/v1/worker/workspace.py:175 的 torch.accelerator.empty_cache()——vLLM 将两个 API 绑到同一 noop 对象，Dynamo 按对象身份分派，导致 empty_cache 被误判为 synchronize。

实现拆解

1. 根因确认：在 Issue 讨论中通过对照实验证明，只要给 empty_cache 一个独立的 no-op，崩溃即消失，不需要修改 Dynamo。
2. 主补丁改动（vllm/v1/worker/cpu/shm.py）：新增模块级函数 empty_cache_noop，并将 torch.accelerator.empty_cache = noop 改为绑定到 empty_cache_noop，与 torch.accelerator.synchronize = noop 解绑。
3. 编译路径补丁改动（vllm/v1/worker/cpu_model_runner.py）：在 _set_torch_accelerator_to_noop 内部同步新增 empty_cache_noop 并替换 empty_cache 绑定，保证两处 CPU 补丁行为一致（该函数是 vLLM 编译配置上下文的组成部分）。
4. 验证与配套：未新增专门测试文件；PR 通过 Buildkite CI（#81959），复现依赖 CPU-only torch 构建（如 torch==2.13.0+cpu）与冷编译缓存，作者提供了最小复现脚本证明修复有效。

关键文件：

- vllm/v1/worker/cpu/shm.py（模块 CPU 后端；类别 source；类型 core-logic；符号 empty_cache_noop）：CPU worker 启动时对 torch 加速器 API 打补丁的核心文件，empty_cache 与 synchronize 的别名在此建立，也是本次修复的主落点
- vllm/v1/worker/cpu_model_runner.py（模块 CPU 后端；类别 source；类型 data-contract；符号 empty_cache_noop, _set_torch_accelerator_to_noop）：编译设置上下文中同样对 torch.accelerator 打补丁，避免 CPU compile 路径下 Dynamo 误判

关键符号：empty_cache_noop, _set_torch_accelerator_to_noop

关键源码片段

vllm/v1/worker/cpu/shm.py

CPU worker 启动时对 torch 加速器 API 打补丁的核心文件，empty_cache 与 synchronize 的别名在此建立，也是本次修复的主落点

```
# CPU-only 主机上对 torch 加速器 API 的统一兜底补丁。
# 两个 no-op 必须保持独立对象，否则 Dynamo 的 handle_synchronize
# 会因共享 callable 对象而把追踪到的 empty_cache() 误判为
# synchronize()，导致 torch.compile 在 aot_compile 阶段断言失败。

def noop(*args: Any, **kwargs: Any) -> None:
    pass


# 独立的 empty_cache no-op：与 synchronize 分开，避免 Dynamo 误判。
def empty_cache_noop(*args: Any, **kwargs: Any) -> None:
    pass


# 统一替换 torch 加速器 API，所有 CPU-only 路径共享这份兜底实现。
torch.accelerator.synchronize = noop
torch.accelerator.empty_cache = empty_cache_noop
```

vllm/v1/worker/cpu_model_runner.py

编译设置上下文中同样对 torch.accelerator 打补丁，避免 CPU compile 路径下 Dynamo 误判

```
def _set_torch_accelerator_to_noop() -> None:
    def noop(*args: Any, **kwargs: Any) -> None:
        pass

    # 关键点：empty_cache 必须使用与 synchronize 不同的 no-op 对象。
    # Dynamo 的 handle_synchronize 以被追踪的 callable 对象为 key，
    # 若两者共用同一 noop，torch.compile 下被追踪的 empty_cache()
    # 会被误分派到 synchronize 处理器，在 CPU-only 主机上触发
    # “No accelerator available for torch.accelerator.synchronize” 断言。
    def empty_cache_noop(*args: Any, **kwargs: Any) -> None:
        pass
```

```
torch.accelerator.synchronize = noop
torch.accelerator.empty_cache = empty_cache_noop
```

评论区精华

> bigPYJ1151：Could you provide a example to reproduce the issue? Tried the openai/gpt-oss-20b with the latest main, cant reproduce the issue on both of MRV1 and MRV2.
>
> ganeshr10：最小复现需要无加速器主机 + 冷编译缓存，GPU 可见的机器永远不会触发；并给出 torch==2.13.0+cpu 下 @torch.compile 内调用 torch.accelerator.synchronize 即断言的示例。
>
> ganeshr10：定位结果——被追踪的调用是 torch.accelerator.empty_cache()（workspace.py:175），不是 synchronize；因为 vLLM 把两者绑到同一 noop 对象，Dynamo 的 handle_synchronize 按对象身份分派，所以 empty_cache 被误判。
>
> bigPYJ1151：Let's add such monkey patches in vllm/v1/worker/cpu/shm.py（随后批准合并）。

- 复现条件与最小示例 (question): 复现依赖 CPU-only torch 构建（如 torch==2.13.0+cpu）与冷编译缓存，GPU 可见的机器不会触发
- 根因定位：共享 noop 对象触发 Dynamo synchronize 断言 (correctness): 已确认根因并采用“独立 empty_cache no-op”方案修复
- 修复落点选择 (design): 在 CPU worker 模块的 torch API 补丁处实现，未改动 Dynamo
- Arm CPU 平台无法复现 (question): 未产生代码改动，最终由 bigPYJ1151 批准合并（根因依赖 CPU-only torch 构建，可能受平台 torch 构建方式影响）

风险与影响

- 风险：依赖 Dynamo 内部实现：修复依据是 Dynamo 以 callable 对象为 key 识别 synchronize 调用；未来 torch 若改变识别方式，补丁可能多余或失效，失效时只是回到崩溃态，不会静默出错。monkey patch 全局性：两处都在 import 阶段改写 torch.accelerator 模块属性，全局生效；但 CPU 上 empty_cache 本就是近似 no-op，语义风险极低。测试覆盖缺口：未新增回归测试，后续重构可能重新引入共享 noop 的问题。影响面：仅 CPU-only 主机、默认 torch.compile 路径的用户受益，GPU / ROCm 路径完全不受影响。
- 影响：对用户：CPU-only 主机上使用默认 torch.compile（非 enforce_eager）的用户不再遇到 aot_compile 崩溃，gpt-oss 等模型冷编译缓存场景可正常工作；eager 模式下同步调用 raise 的问题也一并规避。对系统：CPU worker 启动阶段多一个独立 no-op 对象，无性能开销；语义上 empty_cache 与 synchronize 在 CPU 上仍是空操作。对团队：为 CPU 后端的 Dynamo 兼容性维护提供了可复用的模式，后续若出现类似 API 共享对象被 Dynamo 特殊处理的问题可参考。
- 风险标记：缺少测试覆盖，依赖 Dynamo 内部实现，monkey patch 全局生效

关联脉络

- PR #50547 cpu_model_runner.py: skip the warm up if CompilationMode.NONE: 同改 vllm/v1/worker/cpu_model_runner.py，同属 CPU 后端编译与启动路径的改进
- PR #50133 [CPU] Migrate unquantized MoE to the modular-kernel experts structure: CPU 后端 MoE 模块化重构，与 CPU worker 的 workspace 运行路径存在关联
- PR #50801 [CPU] Refine CPU kernel dispatch: 同属 CPU 后端演进主线，持续收拢 CPU worker 的内核与补丁行为