

PR #50607 完整报告

vllm-project/vllm

[ROCm]: Bump torch 2.12, triton 3.7, torchaudio, torchvision

合并时间: 2026-08-05 09:14

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50607>

执行摘要

- 一句话: ROCm 依赖升级 torch 2.12/triton 3.7, 修复 AOT 加载死锁
- 推荐动作: 值得 ROCm 平台及编译缓存相关开发者精读。本 PR 揭示了一个细致且典型的问题: 依赖升级暴露默认路径的并发加载死锁, 修复方案是平台相关的串行化, 且注释清晰说明了根因。可作为依赖升级与平台 bugfix 组合的范例。

功能与动机

PR body 未给出细节, 但从提交历史看, 目标是同步 ROCm 构建到更新的 PyTorch 版本, 以获取包括 torch.cuda._lazy_call 重入死锁修复在内的更新。作者在 commit 中明确说明, torch 2.12 默认开启 mega-AOT artifact 路径, 该路径在 ROCm 上会导致 AOT 工件反序列化时并发加载 Triton 内核并触发 hipModuleLoad 死锁, 因此需要针对 ROCm 做串行化处理。

实现拆解

1. 在 docker/Dockerfile.rocm_base 中升级依赖: PYTORCH_BRANCH 从 d0c8b1f3 (2.11) 改为 6bbd260 (2.12), TRITON_BRANCH 从 0f380657 改为 0263a6a (3.7.1), torchvision 0.24.1 → 0.27.1, torchaudio 2.9.0 → 2.11.0; 同时引入 gcc-13 (含 update-alternatives 配置), 以提供 torch 2.12 所需的 __truncsfbf2 libcall, 并移除了 triton 上的 cherry-pick 555d04f。
2. 在 vllm/compilation/caching.py 中新增 from vllm.platforms import current_platform 导入, 修改 load_all(): 当 current_platform.is_rocm() 为真时, 串行调用 _load_entry() 反序列化 AOT 工件; 其他平台保留原 ThreadPoolExecutor 并发路径。原因是 ROCm 下反序列化会调用 hipModuleLoad, 并发加载会与 rocprofiler-sdk 的 dl_iterate_phdr 调用产生动态链接器锁死锁。
3. 提交历史显示作者对 torch pin 进行了迭代: 先 pin 到 e4d43db, 随后滚动到包含 ROCm/pytorch#3495 修复的 b10dc69ee (修复 torch.cuda._lazy_call 的 reentrant 死锁, 该问题会导致 vLLM CI 在 torch.compile warmup 阶段挂起)。
4. 测试与 CI 配套: 没有新增单元测试, 改动通过 ROCm CI 验证; CI 中出现两个测试组的回归, 维护者表示将在 PyTorch ticket 中跟踪, 并计划再提两个 PR 修复。

关键文件:

- vllm/compilation/caching.py (模块 编译缓存; 类别 source; 类型 core-logic; 符号 load_all, _load_entry): 核心源码改动: 在 ROCm 平台将 AOT 编译工件加载从并发改为

串行，修复 hipModuleLoad 并发死锁。

- docker/Dockerfile.rocm_base (模块 镜像构建; 类别 infra; 类型 dependency-upgrade) :
基础设施核心: 升级 torch/triton/torchaudio/torchvision 版本, 并引入 gcc-13 以支持 torch 2.12 的编译要求。

关键符号: load_all, _load_entry

关键源码片段

vllm/compilation/caching.py

核心源码改动: 在 ROCm 平台将 AOT 编译工件加载从并发改为串行, 修复 hipModuleLoad 并发死锁。

```
# 文件顶部新增的导入, 用于判断当前运行平台
from vllm.platforms import current_platform
```

```
def load_all(self) -> None:
    import concurrent.futures
```

```
# 如果 submodule store 已经全部加载完成, 直接返回, 避免重复反序列化
if len(self.loaded_submodule_store) == len(self.submodule_bytes_store):
    return
```

```
from torch._inductor.standalone_compile import AOTCompiledArtifact
```

```
def _load_entry(entry_bytes: bytes) -> AOTCompiledArtifact:
    entry = pickle.loads(entry_bytes)
    compilation_counter.num_compiled_artifacts_loaded += 1
    return AOTCompiledArtifact.deserialize(entry)
```

```
entries = list(self.submodule_bytes_store.values())
```

```
if current_platform.is_rocm():
```

```
    # ROCm 平台需要串行加载: 反序列化工件时会触发 `hipModuleLoad`,
    # 并发加载会在 glibc 的动态链接器锁上死锁, 因为 rocprofiler-sdk
    # 在加载过程中会调用 `dl_iterate_phdr`。
```

```
    loaded_entries = [_load_entry(entry) for entry in entries]
```

```
else:
```

```
    # 非 ROCm 平台保持并发加载, 以缩短启动时间
```

```
    with concurrent.futures.ThreadPoolExecutor() as executor:
        loaded_entries = list(executor.map(_load_entry, entries))
```

```
for i, k in enumerate(self.submodule_bytes_store.keys()):
    self.loaded_submodule_store[k] = loaded_entries[i]
```

```
logger.debug("loaded all %s submodules", self.num_artifacts())
```

评论区精华

AndreasKaratzas 在 Changes Requested 审核中列出多条 Buildkite 失败链接，指出 torch 2.12 存在若干问题；随后在 Issue 评论中说明两个回归将在 PyTorch ticket 跟踪，并会有两个 PR 解决对应测试组回归；最终给予 LGTM 批准。作者 Rohan138 则提醒审阅者不要从 main 合并，因为当前 diff 是与昨晚 nightly CI commit 的对比。

- torch 2.12 引入多个 CI 回归 (testing): 后续 AndreasKaratzas 批准 PR (LGTM)，并在评论中说明两个测试组的回归会在 PyTorch ticket 中跟踪，计划再提两个 PR 修复。
- 作者提醒不要从 main 合并 (other): PR 最终合并，未出现冲突问题。

风险与影响

- 风险：
 1. 依赖升级引入两个测试组的回归（见 reviewer 提供的 Buildkite 链接），存在模型或内核层面的兼容性问题。
 2. vllm/compilation/caching.py 将 ROCm 上的 AOT 工件加载从并发改为串行，可能增加启动时间，尤其是大型模型的 mega-AOT artifact 包含大量 submodule 时。
 3. Dockerfile 引入 gcc-13 并切换默认 gcc，可能影响其他使用系统编译器的组件。
 4. 改动仅针对 ROCm 分支，CUDA 等平台不受影响。 - 影响：影响所有使用官方 ROCm 镜像的用户：升级后默认 torch 为 2.12、triton 为 3.7.1，AOT 编译缓存在 ROCm 上的加载行为由并发变为串行（更稳定但可能更慢）。对 CI 而言，ROCM 测试矩阵会暴露回归，目前已知有两个测试组失败，需后续修复。 - 风险标记：ROCM 依赖升级回归（两个测试组），AOT 工件加载串行化可能降低启动性能，gcc-13 切换影响工具链，仅 ROCm 平台受影响

关联脉络

- 暂无明显关联 PR