

# PR #3192 完整报告

radixark/miles

fix(docker): bump torch\_memory\_saver to pick up the free-while-paused fix

合并时间: 2026-09-15 05:55

原文链接: <http://prhub.com.cn/radixark/miles/pull/3192>

## 执行摘要

- 一句话: CUDA 镜像升级 torch\_memory\_saver, 修复暂停态释放崩溃
- 推荐动作: 代码本身只有一行 pin 变更, 不值得逐行精读; 真正值得读的是评论区与 #3243 的验证链条——它解释了为什么「fork PR 的 GPU 红灯不能当证据」, 以及如何用仓内分支承载镜像验证, 对任何改 Dockerfile/ 依赖 pin 的人都直接可复用。关注三个设计决策: 选上游修复的 merge commit (最小 bump) 而不是跟最新版本; CUDA/ROCm 双 pin 的对称性维护; 以及遗留的 segfault 与 backward OOM 必须另开 issue, 不能随本 PR 一起关掉。

## 功能与动机

PR body 给出了完整的回归链路: test\_deepseek\_v4\_flash\_4layer\_ci.py 自 09-09 起在 nightly 报 `[torch_memory_saver.cpp] CUresult error: 1 (invalid argument) file=csrc/core.cpp func=free`, 而「The failure started after weight sync began calling `torch.cuda.empty_cache()` while the trainer is paused by `torch_memory_saver`」——当前 pin `f05a8754d` 在 `empty_cache()` 释放暂停区域内的缓存段时, 会让 `torch_memory_saver` 再次 release 同一 allocation 而崩溃。上游 issue #92 (Freeing a paused allocation double-releases the VMM handle) 指出 `pause()` 已经做过 `cuMemUnmap/cuMemRelease` 并把 allocation 标记为 PAUSED, `free()` 又对同一 handle 再来一遍, 属未定义行为并可能 abort; 该问题由上 PR#95 修复, 本 PR 就是把这个修复引入 CUDA 镜像。

## 实现拆解

1. 定位回归入口: nightly 的 tests/e2e/megatron/model\_scripts/test\_deepseek\_v4\_flash\_4layer\_ci.py 在 torch\_memory\_saver 的 free 路径上抛 `CUresult error: 1 (invalid argument)`, 时间点与 main 上「权重同步期间调用 `torch.cuda.empty_cache()`」的改动吻合, 等价于上游 issue #92 的 double-release 场景。
2. 改一行 pin (唯一源码改动): docker/Dockerfile 中安装 torch\_memory\_saver 的 `RUN TMS_CUDA_MAJOR=$(...) pip install git+https://github.com/fzyzcjy/torch_memory_saver.git@<sha> --no-cache-dir --force-reinstall`, pin 由 `f05a8754daf68238d54e4cf31cb3ba866684bbaf` 换成 `b5588e83de86412a48689a6583a4b567e75f7acc`; TMS\_CUDA\_MAJOR 的推导逻辑、`--no-cache-dir --force-reinstall` 语义都保持不变, 因此镜像构建方式零改动。

3. 选择「最小 bump」而非跟最新：新 sha 是上游 PR#95 的 merge commit，是包含该修复的最小提交；package 版本号仍是 0.0.9.post1，不需要同步改任何依赖约束或编译参数，把引入额外上游变更的风险压到最低。
4. 保持 CUDA/ROCm 对称：ROCm 镜像已 pin 06caa5348，本身就包含该修复，所以本次只动 CUDA 侧，避免两侧行为分叉。
5. 验证与提交结构：本 PR 无单测可加（纯依赖 pin），验证走 GPU 套件——由于 PR 来自 fork，docker-decide 拒绝发布 pr-N 镜像，三条 PR Test 都只能跑在仍带旧 pin 的 radixark/miles:dev 上，于是同 commit 被 cherry-pick 到仓内 PR #3243，在 radixark/miles:pr-3243 镜像上完成实证（test\_deepseek\_v4\_flash\_4layer\_ci.py 转 10/10 通过，CUresult error: 1 计数归零）。提交层面只有 ef7ac9165（pin bump）与 ebd21b13（merge main）两个 commit，属最小化维护型变更。

关键文件：

- docker/Dockerfile（模块 容器镜像；类别 infra；类型 infrastructure；符号 torch\_memory\_saver pin, TMS\_CUDA\_MAJOR）：本 PR 唯一的改动文件：torch\_memory\_saver 的 git pin 由 f05a8754d 换成 b5588e83d，直接决定 CUDA 镜像里是否包含 pause/free double-release 修复，也是 nightly 三个 h200 shard 变红的根因开关。

关键符号：未识别

## 关键源码片段

### docker/Dockerfile

本 PR 唯一的改动文件：torch\_memory\_saver 的 git pin 由 f05a8754d 换成 b5588e83d，直接决定 CUDA 镜像里是否包含 pause/free double-release 修复，也是 nightly 三个 h200 shard 变红的根因开关。

```
# torch_memory_saver 从源码构建，需要 TMS_CUDA_MAJOR 环境变量；
# 这里从 torch 自身的 CUDA 版本推导出主版本号，逻辑本次未改。
#
# 关键改动只有 pin 本身：f05a8754d -> b5588e83d。
# b5588e83d 是上游 PR#95 的 merge commit，修掉的问题是：
# pause() 已经对 allocation 做了 cuMemUnmap / cuMemRelease 并标记 PAUSED，
# 而 free() 又对同一个 VMM handle 再释放一次（double-release，未定义行为 / abort）。
# 触发场景：trainer 被 torch_memory_saver 暂停时，权重同步调用 torch.cuda.empty_cache()，
# 导致暂停区域内的缓存段被 free，最终报 CUresult error: 1 (invalid argument) file=csrc/core.cpp
func=free。
# 之所以只 bump 到这里而不是跟最新版本，是因为它是包含该修复的最小提交，
# package 版本仍是 0.0.9.post1，无需调整任何构建参数或依赖约束。
RUN TMS_CUDA_MAJOR=$(python3 -c "import torch; print(torch.version.cuda.split('.')[0])") \
    pip install git+https://github.com/fzyzcjy/torch_memory_saver.
    git@b5588e83de86412a48689a6583a4b567e75f7acc --no-cache-dir --force-reinstall

# 以下依赖的 pin 本次未改动，列出用于对照：改动是局部的一行。
RUN pip install "nvidia-modelopt[torch]>=0.37.0" --no-build-isolation
RUN pip install git+https://github.com/radixark/Megatron-Bridge.git@bridge --no-deps --no-build-
```

isolation

RUN pip install megatron-energon --no-deps

# 注：ROCm 镜像侧的 pin (06caa5348) 已包含同一修复，因此本 PR 不涉及它，CUDA / ROCm 行为保持一致。

## 评论区精华

讨论几乎全部集中在「这个修复到底有没有被验证」上，而不是代码本身：

yueming-yuan: "The three red NVIDIA shards here are the bug this PR fixes, not evidence against it. This PR comes from a fork, and `docker-decide` in `_build-pr-ci-image.yml` refuses to publish a `pr-N` image for forks ... All three PR Test runs on this branch therefore executed the GPU suites on `radixark/miles:dev`, which still pins `f05a8754`."

这条维护者评论澄清了三条红 shard 的成因是 fork 无法拿到 secrets 发布镜像，而非修复无效，并列出三个受害用例：`test_deepseek_v4_flash_4layer_ci.py`、`test_deepseek_v4_flash_4layer_megatron_impl_ci.py`、`test_qwen3_4B_offload_disaggregated.py`。

yueming-yuan: "#3243 (same commit, in-repo so the image actually builds) has run the GPU suites on `radixark/miles:pr-3243` ... Zero runtime `CUresult error: 1` across every h200 shard that has finished so far; the passing shards exercised 21 weight updates between them. The bump does what it says."

同时给出了边界结论：`test_qwen3_4B_offload_disaggregated.py` 仍挂在 `Reloading 21 process groups → Segfault encountered`，`test_deepseek_v4_flash_4layer_megatron_impl_ci.py` 仍在 backward OOM (6.93 GiB wanted / 8.00 GiB free)，二者在新 pin 下 `CUresult error: 1` 已归零，属独立 bug。另有一条 `claude[bot]` 的系统提示：fork PR 会禁用自动 review，需维护者手动 `@claude review`；最终由 yueming-yuan APPROVED。

- fork PR 的 GPU 红灯是否说明修复无效 (testing): 本 PR 自身 CI 不构成验收证据；改由同 commit 的仓内 PR #3243 构建 `radixark/miles:pr-3243` 镜像完成验证。
- 新 pin 下的实测结果与残留失败归因 (correctness): bump 达成预期；两个残留失败确认为独立 bug (不复现 `CUresult error: 1`)，需要单独跟进，否则 nightly 仍会红并被误读为本 PR 无效。
- fork PR 的自动 review 被禁用 (other): 属于流程性提示而非技术争议，本次改动仅一行 pin，人工批准即可。

## 风险与影响

- 风险：
  1. 供应链与可复现性：pin 指向上游仓库的裸 commit (非 release tag)，若上游 force-push、删库或网络不可达，镜像构建会直接失败；这是既有模式的延续，但本次进一步加深了对该 fork 仓库的隐式依赖。

2. 同版本不同行为：包版本仍为 0.0.9.post1，单看 pip 版本无法判断镜像里是否含修复，排障时必须比对 Dockerfile 中的 sha (b5588e83d vs f05a8754d)，容易被误判为「版本一样所以行为一样」。
  3. 镜像生效存在延迟：Dockerfile 改动只影响之后重建的镜像，radixark/miles:dev 等既有 tag 不会自动更新，nightly 真正转绿取决于镜像发布流水线，存在窗口期内的持续红灯。
  4. 本 PR 自身 CI 不可作为验收证据：fork 分支无法发布 pr-N 镜像，三条 GPU shard 全部跑在旧 pin 的 dev 镜像上，谁只看本 PR 的 checks 会得出反直觉结论；rebuild-ci-image 标签也救不了，因为 fork 检查先于标签读取。
  5. 遗留失败会被误归因：test\_qwen3\_4B\_offload\_disaggregated.py (Reloading 21 process groups 后 segfault) 与 test\_deepseek\_v4\_flash\_4layer\_megatron\_impl\_ci.py (backward OOM) 在新 pin 下依然红，若不单独建 issue 跟踪，nightly 继续红会被再次误认为本次 bump 没生效。
  6. 测试覆盖：纯依赖 pin 无法用单测覆盖，回归保护完全依赖 e2e/nightly，缺少在 CPU 侧快速复现 double-release 的轻量测试。
- 影响：
    - 对 CI/nightly：这是本次变更的主战场。所有 GPU shard 的 CUsresult error: 1 归零，test\_deepseek\_v4\_flash\_4layer\_ci.py 由 FAIL 转 10/10 pass，解除自 09-09 起的持续性红；影响面覆盖 CUDA 镜像承载的全部 GPU 套件 (h200 等)。
    - 对训练 /rollout 用户：无需改配置、无需改代码，重建镜像即可获得修复；被修复的是「trainer 被 torch\_memory\_saver 暂停时，权重同步调用 torch.cuda.empty\_cache()」这条特定路径，属于恢复原有语义而非新增能力。
    - 对 ROCm 用户：无变化，ROCm pin 06caa5348 本就包含修复，两侧行为保持一致。
    - 对团队流程：暴露出 fork PR + 镜像构建的流程盲区，推动形成「同 commit cherry-pick 到仓内分支做镜像级验证」的固定手法 (#3243 的模式)，对后续所有涉及镜像的环境类修复都有复用价值。
    - 风险标记：依赖 pin 到上游未打 tag 的 commit, fork PR 无法自验证镜像，同 package 版本不同行为，遗留 segfault 与 OOM 未修复，回归保护仅依赖 e2e/nightly

## 关联脉络

- PR #3243 同 commit 的仓内分支版本（材料未给出标题）：在评论中被明确引用：把本 PR 的 ef7ac9165 cherry-pick（保留作者）到仓内分支，使 docker-decide 真正构建 radixark/miles:pr-3243 并跑 GPU 套件，是本次修复的唯一有效验证载体。
- PR #3128 fix: use host weights when syncing an offloaded actor: 同属卸载 / 权重同步路径，且与本 PR 中变红的 tests/e2e/megatron/test\_qwen3\_4B\_offload\_disaggregated.py 直接相关；PR body 提到的「weight sync 开始调用 torch.cuda.empty\_cache()」很可能来自这类改动，值得一并回看时间线（材料未点名具体引入 PR，此处为关联推断）。
- PR #2701 feat(ci): declare supported GPU generations on every CUDA test: 同一条 CI 基建脉络：CUDA 测试的 hardware 声明与镜像调度机制，正是本 PR 中「fork 分支跑在 dev 镜像上」这一现象的制度前提，后续硬件感知调度也会影响镜像选择。