PR #44825 完整报告

vllm-project/vllm

[Platform] Replace `torch.cuda.mem_get_info` with `torch.accelerator.get_memory_info`

合并时间：2026-06-30 14:39

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44825

执行摘要

- 一句话：替换 torch.cuda.mem_get_info 为 torch.accelerator.get_memory_info
- 推荐动作：强烈推荐阅读此 PR，尤其是 shm.py 中的 CPU 兼容性补丁设计，以及如何平衡迁移进度与兼容性。后续应继续替换其他 backend 的 mem_get_info，并增加跨平台测试。该 PR 也展示了在 torch.accelerator 原生支持不足时，通过运行时补丁优雅降级的模式。

功能与动机

参考 RFC Issue #30679，vLLM 需要跨硬件后端支持，而 torch.cuda 的硬编码阻碍了非 CUDA 设备（如 Intel GPU、CPU、AMD GPU）的兼容性。PyTorch 提供了 torch.accelerator 抽象 API，可在不同平台上自动调度。本 PR 专注于替换最常用的内存查询 API，作为迁移计划的关键一步。

实现拆解

1. GPU 模型运行器（vllm/v1/worker/gpu_model_runner.py、vllm/v1/worker/gpu/model_runner.py）：将 CUDA graph profiling 中所有 torch.cuda.mem_get_info() 替换为 torch.accelerator.get_memory_info()，用于估算 graph 捕获前后的空闲内存差值以计算内存占用。
2. GPU Worker 核心（vllm/v1/worker/gpu_worker.py、vllm/v1/worker/gpu/spec_decode/eagle/utils.py）：替换 sleep 阶段的空闲内存检测、available memory 判断和 spec decode 内存预算中的 current_platform.mem_get_info() 和 torch.cuda.mem_get_info() 调用。
3. 多模态模型（vllm/model_executor/models/gemma4_mm.py）：替换图像 / 视频编码 chunk 计算中的 current_platform.mem_get_info()，并移除不再需要的 from vllm.platforms import current_platform。
4. CPU 兼容性补丁（vllm/v1/worker/cpu/shm.py）：定义 get_memory_info 函数（封装 get_memory_node_info），并 monkey-patch 到 torch.accelerator.get_memory_info，使 CPU 环境能通过统一 API 查询内存。
5. 平台类清理（vllm/platforms/cpu.py）：删除 CpuPlatform 中的 mem_get_info 类方法（已无调用方），其他平台（Nvidia、AMD）的 mem_get_info 保留以维持向后兼容。
6. 测试与 Lint 更新：更新 tests/basic_correctness/test_mem.py 和 tests/utils_/test_mem_utils.py 使用新 API；修改

tools/pre_commit/check_torch_cuda.py 允许 torch.accelerator 相关调用。

7. 其他适配：删除 vllm/v1/worker/xpu_model_runner.py 中多余的 current_platform 导入（已无引用）。

关键文件：

- vllm/v1/worker/cpu/shm.py（模块 CPU 适配；类别 source；类型 core-logic；符号 get_memory_info）：提供 CPU 环境下 torch.accelerator.get_memory_info 的运行时补丁，是多统一 API 能在 CPU 后端正常工作的关键。在 Python 层面注入 get_memory_info 函数并 patch 到 torch.accelerator，使 rest 代码无需区分后端。
- vllm/v1/worker/gpu_model_runner.py（模块 GPU 运行器；类别 source；类型 data-contract；符号 profile_cudagraph_memory）：CUDA graph 内存 profiling 的核心路径，所有 torch.cuda.mem_get_info 调用被替换。此文件中的修改直接影响了 graph 内存估算的准确性。
- vllm/platforms/cpu.py（模块 平台抽象；类别 source；类型 core-logic；符号 mem_get_info）：删除了不再使用的 mem_get_info 方法，是平台抽象层清除 torch.cuda 依赖的重要步骤。其他平台（Nvidia、AMD）的类似方法因仍有调用而保留。
- vllm/model_executor/models/gemma4_mm.py（模块 多模态模型；类别 source；类型 data-contract；符号 _process_image_input, _process_video_input）：多模态模型编码预算计算中替换 current_platform.mem_get_info()，影响图像 / 视频编码的批量大小决策。同时移除了不用的 current_platform 导入。
- vllm/v1/worker/gpu/model_runner.py（模块 GPU 运行器；类别 source；类型 data-contract；符号 capture_model）：另一个 GPU model runner 中 capture_model 函数的 CUDA graph 内存计算被替换。
- vllm/v1/worker/gpu_worker.py（模块 GPU Worker；类别 source；类型 core-logic；符号 sleep, determine_available_memory）：GPU Worker 中 sleep 阶段的空闲内存检测和 available memory 判断使用 new API，直接影响 GPU memory 管理策略。
- tests/basic_correctness/test_mem.py（模块 内存测试；类别 test；类型 test-coverage）：内存正确性测试用例更新为使用 torch.accelerator.get_memory_info。
- tests/utils_/test_mem_utils.py（模块 内存工具测试；类别 test；类型 test-coverage）：内存工具测试用例同步更新。
- vllm/v1/worker/xpu_model_runner.py（模块 XPU 运行器；类别 source；类型 data-contract）：移除不再需要的 current_platform 导入，是减少 torch.cuda 依赖的清理工作。
- tools/pre_commit/check_torch_cuda.py（模块 Lint 检查；类别 source；类型 core-logic）：预提交检查工具更新，允许 torch.accelerator 相关的调用，防止未来误用 torch.cuda。

关键符号：get_memory_info, CpuPlatform.mem_get_info (deleted), profile_cudagraph_memory, capture_model, sleep, determine_available_memory, _process_image_input, _process_video_input

关键源码片段

vllm/v1/worker/cpu/shm.py

提供 CPU 环境下 torch.accelerator.get_memory_info 的运行时补丁，是多统一 API 能在 CPU 后端正常工作的关键。在 Python 层面注入 get_memory_info 函数并 patch 到 torch.accelerator，使 rest 代码无需区分后端。

```
# 在 vllm/v1/worker/cpu/shm.py 中，为 CPU 环境注入 torch.accelerator.get_memory_info
from vllm.utils.cpu_resource_utils import get_memory_node_info


def get_memory_info(*args: Any, **kwargs: Any) -> tuple[int, int]:
    meminfo = get_memory_node_info()
    # 返回 (available, total)，语义类似 GPU 的 (free, total)
    return meminfo.available_memory, meminfo.total_memory


# 挂载到 torch.accelerator API 上，使统一调用在 CPU 上生效
torch.accelerator.get_memory_info = get_memory_info
```

vllm/v1/worker/gpu_model_runner.py

CUDA graph 内存 profiling 的核心路径，所有 torch.cuda.mem_get_info 调用被替换。此文件中的修改直接影响了 graph 内存估算的准确性。

```
# profile_cudagraph_memory 方法中，使用 torch.accelerator 替代 torch.cuda 进行空闲内存查询
for i, desc in enumerate(profile_descs):
    # 替换前为 torch.cuda.mem_get_info()[0]
    mem_before = torch.accelerator.get_memory_info()[0]
    self._warmup_and_capture(desc, ...)
    torch.accelerator.synchronize()
    # 替换前为 torch.cuda.mem_get_info()[0]
    free_after = torch.accelerator.get_memory_info()[0]
    mem_samples.append(mem_before - free_after)

# 编码器 CUDA graph 同理
if encoder_cudagraph_manager is not None:
    mem_before = torch.accelerator.get_memory_info()[0]
    encoder_cudagraph_manager.capture(graph_pool=encoder_profiling_pool)
    torch.accelerator.synchronize()
    free_after = torch.accelerator.get_memory_info()[0]
    encoder_memory_estimate = max(mem_before - free_after, 0)
```

评论区精华

审查人 hmellor 在 review 中提出：" 替换 torch.cuda.mem_get_info() 看起来没问题，但如果我们也能替换 current_platform.mem_get_info()，是否应该从 platforms 中删除这个方法以避免混淆？" 作者回复解释道：" 我本来也打算替换 current_platform.mem_get_info()，但 torch.accelerator API 在 CPU 上不工作。我将与 @bigPYJ1151 讨论并重新考虑如何替换。" 最终 PR 仅在 cpu.py 中删除了 mem_get_info，其他平台的实现仍保留，同时通过 shm.py 的补丁解决了 CPU 场景。该讨论反映了多硬件统一 API 迁移中的现实约束。另外，DarkLight1337 和 bigPYJ1151 对 PR 进行了批准。

- 是否删除 current_platform.mem_get_info 以避免混淆 (design): 部分删除：仅在 cpu.py 中移除了 mem_get_info 方法（因无调用），其他平台的 mem_get_info 保留。后续可能通过补丁完全替换。
- CPU 上 torch.accelerator 不可用，如何替代 current_platform.mem_get_info (design): 接受了补丁方案，在 CPU shm 中定义 get_memory_info 并 monkey-patch 到 torch.accelerator。

风险与影响

- 风险:
  - 回归风险：torch.cuda.mem_get_info 和 torch.accelerator.get_memory_info 返回值语义一致均为 (free, total)，但需确认在所有后端的一致行为。CPU 补丁返回 available_memory（可能包含缓存）而非 free，可能影响依赖精确空闲内存值的逻辑（如 GPU worker 的 sleep 回收检测）。
  - ROCm / XPU 兼容性：torch.accelerator.get_memory_info 在这些后端上可能尚未实现，本 PR 未添加相应补丁，可能导致 AttributeError。
  - 部分平台 mem_get_info 残留：NvidiaPlatform 等仍保留 mem_get_info，若代码通过 current_platform.mem_get_info() 调用该残留方法，将导致仍使用旧路径，造成维护混淆。
  - 测试覆盖不足：仅更新了已有测试，未新增跨平台的 get_memory_info 行为测试，无法确保所有后端的正确性。
- 影响:
  - 用户影响：对 GPU 用户透明，底层调用切换后行为不变；CPU 用户（之前可能因 torch.cuda 调用而崩溃）现在通过补丁能够正常运行。
  - 系统影响：减少平台特定 API 硬编码，简化后续新硬件支持（如 Intel GPU、AMD GPU）。
  - 团队影响：开发者应优先使用 torch.accelerator API 替代 torch.cuda；预提交检查已更新以阻止新的 torch.cuda 调用。
  - 影响程度：中等。核心执行路径被修改，但替换机制直接且经过测试。
  - 风险标记：核心路径变更，CPU 补丁语义差异，ROCm/XPU 兼容性未知，部分平台 mem_get_info 残留，测试覆盖不全

关联脉络

- 暂无明显关联 PR