PR #46141 完整报告

vllm-project/vllm

[ROCm][CI] Query total device memory via amdsmi to avoid HIP init

合并时间：2026-06-23 04:12

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46141

执行摘要

- 一句话：避免 ROCm 下 HIP 初始化导致 fork 回退
- 推荐动作：此 PR 是 ROCm 平台的关键修复，建议精读。其设计模式（优先使用原生工具避免副作用、安全降级）值得借鉴。

功能与动机

在 ROCm 上，RocmPlatform.get_device_total_memory() 调用 torch.cuda.get_device_properties() 会在父进程中创建 HIP 上下文，导致 vLLM 的多进程逻辑从 fork 回退到 spawn。spawn 模式下工作进程无法继承父进程中注册的模型（如测试用的 PredictableLlamaForCausalLM），导致隐藏状态提取集成测试失败。

实现拆解

1. 新增 amdsmi 导入：在 vllm/platforms/rocm.py 的 try import 块中增加 AmdSmiMemoryType 和 amdsmi_get_gpu_memory_total。
2. 新增查询函数：定义 _query_total_memory_from_amdsmi(physical_device_id) 函数，使用 @with_amdsmi_context 装饰器初始化 / 关闭 amdsmi，通过 amdsmi_get_gpu_memory_total 获取显存总量（字节），保持与 torch.cuda 返回值一致。
3. 重写 get_device_total_memory：将原有 torch.cuda.get_device_properties(device_id).total_memory 替换为优先尝试 amdsmi 路径；若失败则降级到 torch.cuda 并记录一次性警告。函数首先将 device_id 转换为物理设备 ID，再调用 amdsmi 查询。
4. 测试与验证：在 MI300 平台上验证了 get_device_total_memory() 不再初始化 HIP，且回归测试通过。

关键文件：

- vllm/platforms/rocm.py（模块 平台抽象；类别 source；类型 core-logic；符号 _query_total_memory_from_amdsmi, get_device_total_memory）：核心变更文件，新增 amdsmi 查询函数并修改 get_device_total_memory 方法。

关键符号：_query_total_memory_from_amdsmi, get_device_total_memory

关键源码片段

vllm/platforms/rocm.py

核心变更文件，新增 amdsmi 查询函数并修改 get_device_total_memory 方法。

```
# vllm/platforms/rocm.py ( 关键片段 )

# 在 amdsmi 导入块中增加所需符号
try:
    from amdsmi import (
        AmdSmiException,
        AmdSmiMemoryType, # 新增：用于指定 VRAM 类型
        amdsmi_get_gpu_asic_info,
        amdsmi_get_gpu_device_uuid,
        amdsmi_get_gpu_memory_total, # 新增：查询显存总量
        amdsmi_get_processor_handles,
        amdsmi_init,
        amdsmi_shut_down,
        amdsmi_topo_get_link_type,
        amdsmi_topo_get_numa_node_number,
    )
except ImportError as e:
    logger.warning("Failed to import from amdsmi with %r", e)


# 新增函数：通过 amdsmi 查询显存总量，避免初始化 HIP 上下文
@with_amdsmi_context
def _query_total_memory_from_amdsmi(physical_device_id: int) -> int:
    """Query total VRAM (bytes) from amdsmi. Raises if not available."""
    handles = amdsmi_get_processor_handles()
    handle = handles[physical_device_id]
    return amdsmi_get_gpu_memory_total(handle, AmdSmiMemoryType.VRAM)


# 修改后的 get_device_total_memory 方法
@classmethod
def get_device_total_memory(cls, device_id: int = 0) -> int:
    # 优先使用 amdsmi 查询，避免 HIP 上下文创建
    # torch.cuda.get_device_properties() 会创建 HIP 上下文，
    # 导致 vLLM 从 fork 回退到 spawn，破坏自定义模型注册继承。
    try:
        physical_device_id = cls.device_id_to_physical_device_id(device_id)
        return _query_total_memory_from_amdsmi(physical_device_id)
    except Exception as e:
        logger.debug("Failed to get total memory via amdsmi: %s", e)
        logger.warning_once(
            "Failed to get total memory via amdsmi, falling back to "
            "torch.cuda. This will initialize CUDA."
        )
    # 安全降级：使用 torch.cuda（与原行为一致，可能初始化 HIP）
    return torch.cuda.get_device_properties(device_id).total_memory
```

评论区精华

PR 获得两位 reviewer 批准（Rohan138 和 AndreasKaratzas），无额外评论讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。若 amdsmi 不可用或失败，会自动降级到原来的 torch.cuda 路径，行为与之前一致。amdsmi 返回值与 torch.cuda 一致（已验证），不会引起显存计算错误。单文件改动，逻辑清晰。
- 影响：影响范围限于 ROCm 平台。修复了因 HIP 初始化导致的多进程 fork 回退问题，使隐藏状态提取测试和需要自定义模型注册的场景在 ROCm 上正常工作。对 CUDA 平台无影响。
- 风险标记：单文件改动，安全降级存在

关联脉络

- 暂无明显关联 PR