PR #51021 完整报告

vllm-project/vllm

[ROCm] Gate Torch FP8 scaled-MM on architecture support

合并时间：2026-08-19 01:54

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51021

执行摘要

- 一句话：ROCm 按架构门控 FP8 scaled-MM，修复 gfx1100 误选
- 推荐动作：值得精读，虽是小改动（单文件 +26/-17），但揭示了 ROCm 上 kernel 能力判定的系统性陷阱：CUDA 数值 capability 与架构原生能力并非一一对应。重点关注两点：一是 _supports_torch_fp8_scaled_mm() 作为统一出口的分层设计；二是 TODO 中预留的 torch.cuda.is_scaled_mm_supported 替换计划，这是未来消除硬件白名单漂移的关键路径。建议后续将作者的手工验证脚本（平台路由矩阵 + gfx1100/gfx1201 正反用例）沉淀进 tests/，防止架构门控再次回归。

功能与动机

PR body 明确指出根因："The existing implementation compares a CUDA-style numeric capability against the SM89 threshold. On ROCm, gfx1100 reports capability 110, so it incorrectly passes this check even though RDNA3 has no native FP8 matrix support and torch._scaled_mm rejects the device at runtime." 即 capability 数值在 CUDA 上能近似表达 SM 世代，但在 RDNA 上"110"并不对应任何 FP8 矩阵能力，导致选错 kernel 后在运行时崩溃。该 PR 镜像 PyTorch 当前的 ROCm 架构门控，并明确不与 #43615 重复——#43615 解锁 RDNA4 路径，但未修复继承自 per-tensor/channel-wise 的资格检查缺口。

实现拆解

该 PR 的变更入口是 vllm/model_executor/kernels/linear/scaled_mm/pytorch.py，围绕 kernel 选型时的能力判定做收敛与修复，分五步展开：

1. 根因定位：旧 TorchFP8ScaledMMLinearKernel.is_supported 用 compute_capability < 89 作为门槛；RowWise 子类叠加了 get_cdna_version() <= 2 and not on_rdna4() 与 capability 94 判断。两者都依赖 CUDA 风格数值，与 ROCm 真实架构能力脱节。
2. 新增 ROCm 架构白名单：_rocm_torch_fp8_scaled_mm_supported() 从 vllm.platforms.rocm 导入 on_gfx942、on_gfx950、on_gfx12x、on_gfx1250 四个谓词，仅 gfx942（MI300）、gfx950（MI350）、RDNA4 系列返回 True，镜像 PyTorch 当前的 ROCm 门控。
3. 统一判定入口：_supports_torch_fp8_scaled_mm() 做平台分派——CPU/XPU 直接放行，非 CUDA-like 平台拒绝，ROCm 走架构白名单，其余 CUDA 平台走 current_platform.supports_fp8()；并在入口处留 TODO，待 PyTorch 基线纳入

torch.cuda.is_scaled_mm_supported 后整体替换，避免硬件代号漂移。

4. 消费方收敛：基类 is_supported 与 RowWiseTorchFP8ScaledMMLinearKernel.is_supported 删除全部数值 capability 比对，只调用 _supports_torch_fp8_scaled_mm()。基类被 PerTensor 等子类继承，一处修改覆盖多种 kernel 变体。被 gate 掉的路径会自然落入 Triton、upcast、emulation 回退，不改变通用 FP8 支持。
5. 测试与验证：本 PR 未新增常驻测试文件。作者用独立脚本 /tmp/validate_vllm_fp8_scaled_mm.py 验证平台路由矩阵（CPU/XPU 放行、CUDA FP8 能力保留、gfx1100 三种变体全部拒绝、gfx1201 全部接受且与 FP32 参考一致），并触发 Buildkite CI #83126 与 #84416。

关键文件：

- vllm/model_executor/kernels/linear/scaled_mm/pytorch.py（模块 量化内核；类别 source；类型 core-logic；符号 _rocm_torch_fp8_scaled_mm_supported, _supports_torch_fp8_scaled_mm, TorchFP8ScaledMMLinearKernel.is_supported, RowWiseTorchFP8ScaledMMLinearKernel.is_supported）：唯一修改文件，是 Torch FP8 scaled-MM kernel 选择的核心入口。旧实现用 CUDA 数值 capability 判定导致 gfx1100 误通过，本 PR 在此新增 ROCm 架构白名单与统一能力判定入口，基类与 RowWise 的 is_supported 均改走该入口。

关键符号：_rocm_torch_fp8_scaled_mm_supported, _supports_torch_fp8_scaled_mm, TorchFP8ScaledMMLinearKernel.is_supported, RowWiseTorchFP8ScaledMMLinearKernel.is_supported

关键源码片段

vllm/model_executor/kernels/linear/scaled_mm/pytorch.py

唯一修改文件，是 Torch FP8 scaled-MM kernel 选择的核心入口。旧实现用 CUDA 数值 capability 判定导致 gfx1100 误通过，本 PR 在此新增 ROCm 架构白名单与统一能力判定入口，基类与 RowWise 的 is_supported 均改走该入口。

```
# TorchFP8ScaledMMLinearKernel.is_supported 改造后的整体实现（其余
# 子类继承基类，因此一并获得新门控）


def _rocm_torch_fp8_scaled_mm_supported() -> bool:
    # gfx1100（RDNA3）没有原生 FP8 矩阵指令，torch._scaled_mm 会在运行时
    # 拒绝该设备；只有 gfx942（MI300）、gfx950（MI350）、gfx12x（RDNA4）
    # 与 gfx1250 被 PyTorch 的 ROCm 门控放行。
    from vllm.platforms.rocm import on_gfx12x, on_gfx942, on_gfx950, on_gfx1250

    return on_gfx942() or on_gfx950() or on_gfx12x() or on_gfx1250()


def _supports_torch_fp8_scaled_mm() -> bool:
    # CPU 与 XPU 走模拟 / 回退路径，始终可用，直接放行。
    if current_platform.is_cpu():
        return True
```

```
    if current_platform.is_xpu():
        return True
    # 非 CUDA-like 平台没有 torch._scaled_mm 算子。
    if not current_platform.is_cuda_alike():
        return False

    # TODO: 待 PyTorch 基线纳入 torch.cuda.is_scaled_mm_supported 后，
    # 用官方查询替换本架构白名单，避免新硬件代号导致门控漂移。
    if current_platform.is_rocm():
        return _rocm_torch_fp8_scaled_mm_supported()
    return current_platform.supports_fp8()


@classmethod
def is_supported(
    cls, compute_capability: int | None = None
) -> tuple[bool, str | None]:
    # 旧实现用 SM89 数值阈值判定，gfx1100 上报 capability 110 会误通过；
    # 改为平台能力判定后，ROCm 按架构族白名单决策，CUDA 仍按 FP8 能力。
    if _supports_torch_fp8_scaled_mm():
        return True, None

    return False, "requires a platform with torch FP8 scaled-MM support."
```

评论区精华

本 PR 没有实质性的 review 争论或 inline comments（review_comments_count = 0）。审核记录仅有 tjtanaa 的 APPROVE，以及 claude[bot] 的自动说明——这是 fork PR，默认不开启自动化审查，维护者可评论 @claude review 手动触发。6 条 issue 评论全部是 /ci run 指令与 GitHub Actions 的构建触发回执，无技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
    1. 架构白名单依赖人工维护：_rocm_torch_fp8_scaled_mm_supported() 依赖 vllm.platforms.rocm 的 on_* 谓词，未来新增 ROCm 架构（如新的 gfx 代号）若不及时加入名单，会在本可支持的硬件上误拒 Torch scaled-MM。PR 已留 TODO，待 torch.cuda.is_scaled_mm_supported 进入最低 PyTorch 版本后替换。
1. channel-wise 覆盖待确认：PR body 声称同时门控 per-tensor、channel-wise、row-wise 三种变体，但 patch 可直接观察到的是基类与 RowWise 两处 is_supported 改动；channel-wise 是否经由继承自动获得门控（还是存在于其他文件）未见显式修改，建议确认。
2. CUDA 路径语义变化：CUDA 从 compute_capability < 89 数值比较改为 current_platform.supports_fp8()，两者语义不完全一致（supports_fp8 还考虑平台整体能力），若某 CUDA 平台该谓词判定变化会影响 kernel 选择，需回归验证。

3. 缺少常驻自动化测试：验证依赖作者本地独立脚本，未沉淀为 tests/ 下的 CI 用例，架构门控未来回归风险较高。
4. 测试覆盖受限：作者注明本地 vLLM 扩展与安装的 PyTorch ABI 不兼容，验证只覆盖了 Python 选型逻辑和底层 PyTorch 算子，未跑完整端到端模型 serving。 - 影响：
    1. 用户面：ROCm（AMD GPU）上运行 FP8 量化模型的用户。gfx1100 等 RDNA3 设备从“选错 kernel 后运行时崩溃”变为“kernel 选择阶段即拒绝 Torch 路径并自动回退到 Triton/upcast/emulation”，可用性显著提升。
5. 系统面：gfx942（MI300）、gfx950（MI350）、RDNA4（gfx12x/gfx1250）继续使用 torch._scaled_mm，性能不变；CPU/XPU/CUDA 行为保留。
6. 团队面：kernel 能力判定收敛为单一辅助函数，为后续接入 torch 官方 is_scaled_mm_supported 提供了清晰的替换点，降低多平台分支维护成本。 - 风险标记：核心路径变更，缺少自动化测试覆盖，硬件白名单依赖人工维护，channel-wise 覆盖待确认

关联脉络

- PR #43615 （标题未提供，见 PR body）：PR body 明确提及该 PR：它解锁 RDNA4 的 FP8 scaled-MM 路径，但没有修复继承自 per-tensor/channel-wise 的资格检查在 gfx1100 上的误判，本 PR 正是补齐该缺口。
- PR #52182 Remove VLLM_TEST_FORCE_FP8_MARLIN to replace with linear_backend/moe_backend: 同属 FP8 线性内核选择与后端抽象演进脉络，都在调整 linear/scaled_mm 周边路径的能力判定与后端路由逻辑。
- PR #52112 [Bugfix][ROCm] Fix a few int4/int8 quantization errors: 同属 ROCm 平台量化内核选择修复，说明 ROCm 上“数值能力判定与实际硬件能力脱节”是一类持续存在的问题面。