PR #41436 完整报告

vllm-project/vllm

[ROCm][Quantization][3/N] Refactor quark_moe w4a4 w/ oracle

合并时间：2026-05-19 01:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41436

执行摘要

- 一句话：重构 quark_moe W4A4 使用 oracle 后端选择机制
- 推荐动作：建议所有关注 ROCm 后端的工程师和量化框架开发者精读。该 PR 展示了如何通过 oracle 模式统一多 backend 选择，并清晰展示了重构渐变过程（3/N）。值得注意的设计决策包括：emulation 的 opt-in 策略；简化初始化流程以消除冗余条件；错误消息的可操作性改进。对于需要支持新硬件后端的工程师，此模式是很好的参考。

功能与动机

PR #41436 旨在完成 quark_moe 所有 MXFP4 量化方案向 oracle 的迁移（W4A16, W4A8 已完成，W4A4 是最后一块）。统一后端选择逻辑减少重复代码，并为 ROCm 用户提供 GFX950 上的原生 W4A4 支持。同时修正原有 emulation 隐式回退的模糊语义，要求用户显式指定 --moe-backend emulation 才能使用模拟路径。

实现拆解

1. 在 quark_moe.py 中添加 W4A4 分支：在 QuarkOCP_MX_MoEMethod.__init__中新增 elif self.ocp_mx_scheme == "w_mxfp4_a_mxfp4" 分支，调用 select_mxfp4_moe_backend(moe, activation_key=kMxfp4Dynamic) 进行后端选择。
2. 在 mxfp4.py oracle 中注册新后端：在 Mxfp4MoeBackend 枚举中添加 AITER_MXFP4_MXFP4；在 backend_to_kernel_cls、AITER_BACKENDS、map_mxfp4_backend、_get_priority_backends_for_gpt_oss 和 _backend_activation_key 等关键函数中注册该后端，确保它被识别、优先级排序和激活键映射正确。
3. 在 rocm_aiter_moe.py 中声明量化方案支持：重写 is_supported_config 方法，在父类不支持的场景下给出具体环境变量指引；在 _supports_quant_scheme 中添加 (kMxfp4Static, kMxfp4Dynamic) 组合，声明对 W4A4 的支持。
4. 简化 quark_moe.py 的后端选择逻辑：移除原有的 self.use_rocm_aiter_moe、self.emulate 等复杂条件变量，统一使用 Mxfp4MoeBackend.NONE 判断；若未选择任何后端则回退至 EMULATION，并通过 backend_to_kernel_cls 获取对应内核类。同时将分散的条件日志简化为一条统一的后端选择信息日志。
5. 新增 / 修改测试和评估配置：将 tests/quantization/test_gfx950_moe.py 从占位 TODO 重写为三个真实测试用例（W4A4 调度到 AITER、无 AITER 时引发错误、显式

EMULATION 后端调度）；添加 tests/evals/gsm8k/configs/Qwen3.5-35B-A3B-MXFP4-AITER-TP2.yaml 用于 AITER 路径的 GSM8K 精度评估；重命名原 Qwen3.5-35B-A3B-MXFP4-TP2.yaml 为 EMU 版本并调整准确率阈值和 server_args。

关键文件：

- vllm/model_executor/layers/quantization/quark/quark_moe.py（模块 量化层；类别 source；类型 data-contract；符号 process_weights_after_loading）：核心重构文件：移除旧的 emulation 逻辑，引入 W4A4 oracle 后端选择，简化权重加载后处理。
- vllm/model_executor/layers/fused_moe/oracle/mxfp4.py（模块 后端选择；类别 source；类型 data-contract）：新增 AITER_MXFP4_MXFP4 后端枚举，注册到所有 oracle 选择结构，处理 W4A4 的后端优先级和过滤。
- tests/quantization/test_gfx950_moe.py（模块 GFX950 测试；类别 test；类型 test-coverage；符号 test_mi355_moe, _make_w4a4_moe_config, test_w4a4_dispatches_to_aiter, test_w4a4_raises_without_aiter_and_no_moe_backend）：从占位 TODO 实现为完整的 W4A4 后端选择测试，覆盖 AITER、无 AITER、EMULATION 三种场景。
- vllm/model_executor/layers/fused_moe/experts/rocm_aiter_moe.py（模块 AITER 专家；类别 source；类型 data-contract；符号 is_supported_config）：为 W4A4 添加 is_supported_config 和 quant_scheme 支持，提供清晰的启用提示。
- tests/evals/gsm8k/configs/Qwen3.5-35B-A3B-MXFP4-AITER-TP2.yaml（模块 评估配置；类别 test；类型 test-coverage）：新增 AITER 后端 W4A4 评估配置，用于 GSM8K 精度验证。
- tests/evals/gsm8k/configs/Qwen3.5-35B-A3B-MXFP4-EMU-TP2.yaml（模块 评估配置；类别 test；类型 rename-or-move）：从原文件重命名，添加 --moe-backend emulation 参数并调整准确率阈值。
- tests/evals/gsm8k/configs/models-mi3xx.txt（模块 模型列表；类别 docs；类型 documentation）：更新 mi3xx 模型列表，记录 Qwen3.5-35B-A3B-MXFP4 评估配置。
- tests/evals/gsm8k/configs/models-qwen35-mi355.txt（模块 模型列表；类别 docs；类型 documentation）：更新 Qwen3.5 mi355 模型列表，记录新配置。

关键符号：process_weights_after_loading, is_supported_config, select_mxfp4_moe_backend, _supports_quant_scheme, backend_to_kernel_cls

关键源码片段

vllm/model_executor/layers/quantization/quark/quark_moe.py

核心重构文件：移除旧的 emulation 逻辑，引入 W4A4 oracle 后端选择，简化权重加载后处理。

```
# 在 QuarkOCP_MX_MoEMethod.__init__ 中，根据 OCP MX 方案选择后端
# 新增 W4A4 分支：ocp_mx_scheme == "w_mxfp4_a_mxfp4"
if self.ocp_mx_scheme == "w_mxfp4":
    # W4A16: weight-only MXFP4
    self.mxfp4_backend, self.experts_cls = select_mxfp4_moe_backend(moe)
```

```
elif self.ocp_mx_scheme == "w_mxfp4_a_fp8" and self.static_input_scales:
    # W4A8: MXFP4 weights + static FP8 activations
    self.mxfp4_backend, self.experts_cls = select_mxfp4_moe_backend(
        moe, activation_key=kFp8StaticTensorSym
    )
elif self.ocp_mx_scheme == "w_mxfp4_a_mxfp4":
    # W4A4: MXFP4 weights + MXFP4 activations
    self.mxfp4_backend, self.experts_cls = select_mxfp4_moe_backend(
        moe, activation_key=kMxfp4Dynamic
    )

# 如果没有原生后端可用，回退到 EMULATION
if self.mxfp4_backend is Mxfp4MoeBackend.NONE:
    self.mxfp4_backend = Mxfp4MoeBackend.EMULATION

# 从后端映射到具体内核类
self.experts_cls = backend_to_kernel_cls(self.mxfp4_backend)[0]
```

vllm/model_executor/layers/fused_moe/experts/rocm_aiter_moe.py

为 W4A4 添加 is_supported_config 和 quant_scheme 支持，提供清晰的启用提示。

```
@staticmethod
def is_supported_config(cls, moe_config, weight_key, activation_key, activation_format):
    # 调用父类检查基础支持性
    is_supported, reason = super().is_supported_config(
        cls, moe_config, weight_key, activation_key, activation_format
    )
    # 如果不支持且 AITER MoE 未启用，给出明确的环境变量指引
    if not is_supported and not rocm_aiter_ops.is_fused_moe_enabled():
        reason = (
            f"{reason}. AITER MoE is not enabled — "
            "set VLLM_ROCM_USE_AITER=1 and VLLM_ROCM_USE_AITER_MOE=1 "
            "to enable it"
        )
    return is_supported, reason

@staticmethod
def _supports_quant_scheme(weight_key, activation_key):
    SUPPORTED_W_A = [
        (None, None),
        (kFp8Static128BlockSym, kFp8Dynamic128Sym),
        (kFp8StaticTensorSym, kFp8StaticTensorSym),
        (kFp8StaticTensorSym, kFp8DynamicTensorSym),
        (kFp8StaticChannelSym, kFp8DynamicTokenSym),
        (kMxfp4Static, None),
        (kMxfp4Static, kMxfp4Dynamic), # 新增：W4A4 MXFP4 weights + MXFP4 activations
    ]
    if (weight_key, activation_key) not in SUPPORTED_W_A:
        return False
```

```
    # CK MXFP4 MoE kernels 仅在 GFX950 硬件上可用
    if weight_key == kMxfp4Static:
        from vllm.platforms.rocm import on_gfx950
        if not on_gfx950():
            return False
    return True
```

评论区精华

> mgoin在 Qwen3.5-35B-A3B-MXFP4-AITER-TP2.yaml 中建议：能否在 server_args 中使用 --moe-backend aiter 而不是仅依赖环境变量？BowenBao同意，并补充仍需保留 VLLM_ROCM_USE_AITER 作为全局开关。
>
> mgoin在 mxfp4.py 中询问：是否有意从 _get_priority_backends_for_gpt_oss 中移除了 EMULATION？BowenBao解释：这是有意设计，使 EMULATION 成为 opt-in 以避免意外性能下降；错误消息已更新提示用户使用 --moe-backend emulation。
>
> gemini-code-assist指出 test_ocp_mx_moe.py 中的 reference_moe 函数和权重反量化逻辑存在代码重复，建议提取辅助函数以提高可维护性。该建议在最终 commit 中未看到明确响应。

- EMULATION 后端从自动优先级移除 (design): 确认移除 EMULATION 是设计决策，用户需显式指定后端。
- AITER 评估配置使用 --moe-backend aiter (question): 在 server_args 中添加了 --moe-backend aiter，保留环境变量。
- 测试代码中 MXFP4 量化 / 反量化逻辑重复 (style): 未在最终 commit 中看到明确修复，可能为遗留问题。

风险与影响

- 风险：
  1. 平台兼容性风险：新增的 W4A4 路径仅针对 GFX950 硬件，非 GFX950 的 ROCm 平台将收到 NotImplementedError，要求显式使用 EMULATION 后端。未改变其他平台的行为。
  2. 配置迁移风险：emulation 后端从自动优先级中移除，依赖隐式 EMULATION 的用户需要显式指定 --moe-backend emulation，否则后端选择将失败。已通过更新错误消息进行缓解。
  3. 测试覆盖风险：test_gfx950_moe.py 的测试仅在 GFX950 真机上运行，CI 中若缺少相应硬件则被跳过，可能遗漏回归。需要定期真实硬件验证或考虑 mock 测试。 - 影响：用户影响：GFX950 用户获得一站式 W4A4 支持，无需手动指定后端；其他 ROCm 用户获得更清晰的错误提示和解决方向；EMULATION 用户需要主动添加 --moe-backend emulation 参数。系统影响：简化了 quark_moe 的初始化流程，减少条件分支和日志噪音，提高可维护性。团队影响：统一 oracle 体系后，新增新量化后端（如其他 W4A4 实现）只需在 oracle 中注册，无需修改 quark_moe 主路径，降低了耦合度。
- 风险标记：核心路径变更，平台特定代码，测试依赖真实硬件，配置参数变更

关联脉络

- PR #39136 [ROCm][Quantization] Refactor quark_moe w4a4 w/ oracle: 此 PR 的基础变更，包含 refactor 的初始代码。