PR #51624 完整报告

vllm-project/vllm

[Hardware][Power] Unqualized MoE Backend for Power (VSX)

合并时间：2026-08-13 10:56

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51624

执行摘要

- 一句话：为 Power 架构添加 VSX 未量化 MoE 后端
- 推荐动作：值得精读，特别是 cpu_micro_gemm_vsx.hpp 中的 MMA 指令使用和优化过程（25 个提交中有大量性能调优）。关注其如何在统一接口下插入架构特定实现，以及如何在测试中启用。

功能与动机

根据 PR body，当前 vLLM 中 grouped GEMM 不支持 Power 架构，因此引入 Power/VSX 特定未量化 CPU 后端，以利用 Power10 MMA 指令提升性能。作者测试表明，相比 main 分支，预填充阶段 TTFT 有 35% 左右的提升。

实现拆解

1. 新增 csrc/cpu/micro_gemm/cpu_micro_gemm_vsx.hpp，实现 TileGemmVSX 模板，基于 Power10 MMA 指令（如 __builtin_mma_xvbf16ger2pp）提供 8-16-16 模式的 GEMM 计算，并包含 M<=4 的 VSX 回退路径。
2. 扩展 csrc/cpu/utils.hpp 中 ISA 枚举，新增 VSX 类型，并支持 "vsx" 字符串解析。
3. 在 csrc/cpu/cpu_fused_moe.cpp 中，通过 __powerpc64__ 条件编译添加 VSX 分发宏，使 CPU fused MoE 能够调用 VSX 微内核。
4. 在 vllm/model_executor/layers/fused_moe/experts/cpu_moe.py 中新增 PowerCPUUnquantizedExperts 类，设置对齐参数并定义设备支持检查，作为后端选择入口。
5. 在 vllm/model_executor/layers/fused_moe/oracle/unquantized.py 中接入 PowerCPUUnquantizedExperts 到后端选择列表。
6. 测试配套：更新 tests/kernels/moe/test_cpu_fused_moe.py，在 ISA 列表和专家类映射中加入 VSX。

关键文件：

- csrc/cpu/micro_gemm/cpu_micro_gemm_vsx.hpp（模块 CPU 内核；类别 source；类型 core-logic；符号 TileGemmVSX, MicroGemm）：核心新增文件，实现 VSX 指令集的 GEMM 微内核，是性能提升的关键

- vllm/model_executor/layers/fused_moe/experts/cpu_moe.py（模块 MoE 专家；类别 source；类型 data-contract；符号 PowerCPUUnquantizedExperts, _supports_current_device, is_supported_config）：新增 PowerCPUUnquantizedExperts 类，定义后端选择逻辑和设备支持检查
- csrc/cpu/cpu_fused_moe.cpp（模块 内核分发；类别 source；类型 dependency-wiring）：在 CPU fused MoE 分发逻辑中接入 VSX 宏，使内核可被调度
- csrc/cpu/utils.hpp（模块 工具层；类别 source；类型 core-logic；符号 ISA）：扩展 ISA 枚举和解析，支持 "vsx" 字符串
- vllm/model_executor/layers/fused_moe/oracle/unquantized.py（模块 后端路由；类别 source；类型 data-contract）：在后端选择器中注册 PowerCPUUnquantizedExperts，确保自动路由
- tests/kernels/moe/test_cpu_fused_moe.py（模块 MoE 测试；类别 test；类型 test-coverage）：更新测试以覆盖 VSX 后端，验证 Power 架构下的正确性

关键符号：TileGemmVSX::gemm, TileGemmVSX::gemm_micro, TileGemmVSX::gemm_micro_vsx_fallback, PowerCPUUnquantizedExperts._supports_current_device, PowerCPUUnquantizedExperts.is_supported_config

关键源码片段

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

新增 PowerCPUUnquantizedExperts 类，定义后端选择逻辑和设备支持检查

```
class PowerCPUUnquantizedExperts(CPUUnquantizedExperts):
    """PowerPC VSX grouped-gemm unquantized MoE experts."""

    # 标识使用的 ISA 类型，用于运行时选择
    isa = "vsx"
    # 输出对齐：16 字节，满足 VSX 向量访问要求
    output_alignment = 16
    # 归约对齐：2 表示 BF16 的字节宽度
    reduction_alignment = 2

    @staticmethod
    def _supports_current_device() -> bool:
        # 仅在 PowerPC 架构的 CPU 上启用此后端
        return (
            current_platform.is_cpu()
            and current_platform.get_cpu_architecture() == CpuArchEnum.POWERPC
        )

    @staticmethod
    def is_supported_config(
        cls: type[mk.FusedMoEExperts],
        moe_config: FusedMoEConfig,
        weight_key: QuantKey | None,
        activation_key: QuantKey | None,
```

```
        activation_format: mk.FusedMoEActivationFormat,
    ) -> tuple[bool, str | None]:
        # 先调用基类的通用检查，确保基本配置合法
        supported, reason = mk.FusedMoEExperts.is_supported_config(
            cls, moe_config, weight_key, activation_key, activation_format
        )
        if not supported:
            return supported, reason
        # 此内核仅支持 bfloat16 激活
        if moe_config.in_dtype != torch.bfloat16:
            return False, "kernel requires bfloat16 activations"
        # 复用 CPU 未量化专家的 grouped-gemm 支持检查
        cpu_cls = cast(type[CPUUnquantizedExperts], cls)
        return cpu_cls._supports_grouped_gemm(moe_config)
```

评论区精华

bigPYJ1151 在 review 中指出，ArmCPUUnquantizedExperts 的 is_supported_config 实现应保留，并为 PowerCPUUnquantizedExperts 重复该逻辑。作者已回复“Addressed this, apologies on my end for this.”并修正。这条评论体现了对基类结构一致性的关注。

- ArmCPUUnquantizedExperts 结构保持 (design): 作者已修正，将 is_supported_config 重复到新的 PowerCPUUnquantizedExperts 类中，保持基类结构不变。

风险与影响

- 风险：新增架构特定代码，但通过条件编译和 _supports_current_device 限制，其他架构不受影响。性能调优复杂（25 个提交中有多次优化与回退），且仅在 Power10 上验证，其他 Power 型号（如 POWER9）可能未覆盖。测试仅在 CPU 架构上运行，Power 架构的 CI 覆盖可能缺失。
- 影响：对 Power 用户的预填充性能提升显著，对非 Power 用户无影响。团队需维护额外的架构特定代码，但通过模块化设计降低了侵入性。
- 风险标记：架构特定实现，缺少 Power CI 验证，性能调优复杂

关联脉络

- 暂无明显关联 PR