PR #43404 完整报告

vllm-project/vllm

[XPU] add awq format for INCXPULinear

合并时间：2026-06-22 22:29

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43404

执行摘要

- 一句话：INCXPULinear 增加 AWQ 权重格式支持
- 推荐动作：值得阅读，特别是量化格式之间的转换技巧。关注 _convert_awq_qweight_to_gptq 的实现方式，可作为不同量化方案互转的参考。建议后续增加针对转换函数的单元测试以验证正确性。

功能与动机

PR body 指出："convert AWQ to GPTQ, letting INCXPULinear can handle AWQ-format autoround models load." 用户需要在 XPU 上使用 INC 量化时加载 AWQ 格式的 AutoRound 量化模型。

实现拆解

1. 格式标记与常量：在 INCXPULinearBase.__init__ 中从 layer_config.is_awq 读取格式标记；定义 _REVERSE_AWQ_PACK_ORDER 常量描述 AWQ 到 GPTQ 的 nibble 反排列顺序。
2. 权重张量创建：在 _create_inc_weights 中根据 is_awq_packed 选择 qweight 的形状和打包维度——AWQ 格式为 [K, N//pack_factor]（输出打包），GPTQ 格式为 [K//pack_factor, N]（输入打包）。其他参数（scales、qzeros、g_idx）的创建不变。
3. 格式转换函数：新增 _convert_awq_qweight_to_gptq，该函数首先反解 AWQ nibble 顺序，然后通过 torch.nn.functional.scatter_ 将 nibble 按 GPTQ 的次序重填入新的 int32 张量中，最终重塑为 [K//pf, N] 形状。该转换保持数值完全无损。
4. 权重后处理调用（推断）：在 INCXPULinearMethod.process_weights_after_loading 中判断若 is_awq_packed，则调用上述转换函数将权重重排为 GPTQ 格式，随后执行原有的转置与参数封装。
5. 测试适配：将 tests/quantization/test_auto_round.py 中 AWQ 模型测试的跳过条件从 not is_cuda() 扩展为 not (is_cuda() or is_xpu())，使 XPU 可参与测试。

关键文件：

- vllm/model_executor/layers/quantization/inc/schemes/inc_wna16_linear.py（模块 INC 量化；类别 source；类型 core-logic；符号 _convert_awq_qweight_to_gptq, is_awq_packed, _REVERSE_AWQ_PACK_ORDER）：核心实现文件，添加了 AWQ 格式检测、参数创建分支和转换函数

- tests/quantization/test_auto_round.py（模块 量化测试；类别 test；类型 test-coverage）：测试配套修改，扩展 XPU 支持

关键符号：_convert_awq_qweight_to_gptq, INCXPULinearBase.init, INCXPULinearBase._create_inc_weights

关键源码片段

vllm/model_executor/layers/quantization/inc/schemes/inc_wna16_linear.py

核心实现文件，添加了 AWQ 格式检测、参数创建分支和转换函数

```
# INCXPULinearBase 中新增的 AWQ 格式支持
class INCXPULinearBase(INCLinearScheme):
    # AWQ 的 nibble 打包顺序为 [0,2,4,6,1,3,5,7]，
    # 该常量定义反向置换以恢复为顺序排列。
    _REVERSE_AWQ_PACK_ORDER = [0, 4, 1, 5, 2, 6, 3, 7]

    def __init__(self, layer_config: 'INCLayerConfig') -> None:
        self.weight_bits = layer_config.bits
        self.group_size = layer_config.group_size
        self.sym = layer_config.sym
        self.pack_factor = 32 // self.weight_bits
        self.is_awq_packed = layer_config.is_awq # AWQ 格式标记

    def _create_inc_weights(self, ...):
        # ... 公共形状计算 ...
        if self.is_awq_packed:
            # AWQ: 沿输出维度打包，形状 [K, N//pf]
            qweight = PackedvLLMParameter(
                data=torch.empty(input_size_per_partition,
                                 output_size_per_partition // self.pack_factor,
                                 dtype=torch.int32),
                input_dim=0, output_dim=1, packed_dim=1,
                packed_factor=self.pack_factor, weight_loader=weight_loader,
            )
        else:
            # GPTQ: 沿输入维度打包，形状 [K//pf, N]
            qweight = PackedvLLMParameter(
                data=torch.empty(input_size_per_partition // self.pack_factor,
                                 output_size_per_partition,
                                 dtype=torch.int32),
                input_dim=0, output_dim=1, packed_dim=0,
                packed_factor=self.pack_factor, weight_loader=weight_loader,
            )
        # scales, qzeros, g_idx 的创建不区分格式，此处省略
```

评论区精华

审阅者 yiliu30 和 jikunshang 均批准该 PR，无重大争议。yiliu30 在评论中表示 'LGTM! Thanks for the support!'，并提及需要适配新的 INC flow（PR #40601）。

- 适配新的 INC flow (*design*): 作者遵循了建议，在 INCXPULinearBase 中通过 *layer_config.is_awq* 判断格式，与新的 INC flow 一致。

风险与影响

- 风险：
  - 数值正确性：转换逻辑理论上无损，但缺乏独立的单元测试覆盖，仅依赖端到端生成测试。建议后续补充 _convert_awq_qweight_to_gptq 的验证。
  - 性能：权重加载时增加一次转换，对运行时推理无影响。
  - 兼容性：只修改 XPU 路径，CUDA/CPU 后端不受影响。
  - 平台风险：新增分支在非 XPU 设备上不执行，但需确保配置 is_awq 在其他平台不会意外启用。
- 影响：
  - 用户影响：XPU 用户现在可以直接加载 AutoRound 量化且使用 AWQ 格式的模型（如 Intel/Qwen2-0.5B-Instruct-int4-sym-AutoRound），无需手动转换权重格式。
  - 系统影响：新增约 90 行代码，未破坏现有 INC 流程。
  - 团队影响：与 #40601 的新 INC 流程衔接，统一了不同打包格式的处理方式。
  - 风险标记：数值精度风险，转换函数未单独测试，仅 XPU 路径验证

关联脉络

- PR #40601 Unknown (*referenced as new INC flow*): yiliu30 要求适配该 PR 中的新 INC 流程，此实现与之对齐。