PR #27533 完整报告

sgl-project/sglang

[Intel GPU] Enable fused_experts in fp8.py for quantized models on XPU

合并时间：2026-06-09 09:22

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27533

执行摘要

- 一句话：XPU 启用 fused_experts 量化支持
- 推荐动作：建议阅读以了解 XPU 支持的扩展方式。后续应补充模型级测试（如 reviewer 提到的 gpt-oss 20B）以验证正确性。

功能与动机

在 Intel XPU 上运行量化模型时，原先的 fused_experts 路径仅针对 CUDA、HIP 和 CPU（AMX）设计，缺少 XPU 特定实现。此 PR 通过引入 sgl-kernel-xpu 的内核支持，使 XPU 也能享受 fused_experts 的性能优势。

实现拆解

1. 在 python/sglang/srt/layers/quantization/fp8.py 中从 sglang.srt.utils 导入 use_intel_xpu_backend 函数。
2. 在 FusedMoEMethod 类的 apply 方法中，在 HIP aiter 分支之后、cutlass 分支之前，新增一条条件分支：当 use_intel_xpu_backend() 返回 True 时，导入并调用 sgl_kernel.fused_experts。
3. 根据权重的 dtype 判断量化模式：torch.float8_e4m3fn 为 FP8 W8A8，torch.int8 为 MXFP4 W4A16，并通过 self.is_fp4_expert 进行断言验证。
4. 调用 fused_experts 时，根据 self.block_quant 选择使用 weight_scale_inv 或 weight_scale 作为缩放因子，并传递 bias、activation、routed_scaling_factor、gemm alpha/limit 和 swiglu limit 等配置参数。

关键文件：

- python/sglang/srt/layers/quantization/fp8.py（模块 量化；类别 source；类型 dependency-wiring；符号 FusedMoEMethod.apply）：唯一变更文件，新增 XPU 分支的 fused_experts 调用逻辑和导入。

关键符号：FusedMoEMethod.apply

关键源码片段

python/sglang/srt/layers/quantization/fp8.py

唯一变更文件，新增 XPU 分支的 fused_experts 调用逻辑和导入。

```
# python/sglang/srt/layers/quantization/fp8.py
# 在 apply 方法中，于 HIP aiter 分支之后、cutlass 分支之前插入 XPU 路径

if use_intel_xpu_backend():
    # 通过 sgl-kernel-xpu 的 fused_experts 内核，在 XPU 上执行 MoE 专家计算
    from sgl_kernel import fused_experts

    topk_weights, topk_ids, _ = dispatch_output.topk_output
    # 根据权重 dtype 判断量化格式
    # float8_e4m3fn → FP8 W8A8，int8 → MXFP4 W4A16
    assert layer.w13_weight.dtype == layer.w2_weight.dtype
    use_fp8_w8a8 = layer.w13_weight.dtype == torch.float8_e4m3fn
    use_mxfp4_w4a16 = layer.w13_weight.dtype == torch.int8
    # 断言 is_fp4_expert 标志与实际的 MXFP4 量化一致
    assert self.is_fp4_expert == use_mxfp4_w4a16

    output = fused_experts(
        x,
        layer.w13_weight,
        layer.w2_weight,
        topk_weights,
        topk_ids,
        b1=getattr(layer, "w13_weight_bias", None),
        b2=getattr(layer, "w2_weight_bias", None),
        use_mxfp4_w4a16=use_mxfp4_w4a16,
        use_fp8_w8a8=use_fp8_w8a8,
        # block_quant 为 True 时使用 weight_scale_inv，否则使用 weight_scale
        w1_scale=(
            layer.w13_weight_scale_inv
            if self.block_quant
            else layer.w13_weight_scale
        ),
        w2_scale=(
            layer.w2_weight_scale_inv
            if self.block_quant
            else layer.w2_weight_scale
        ),
        activation=moe_runner_config.activation,
        routed_scaling_factor=moe_runner_config.routed_scaling_factor,
        gemm1_alpha=moe_runner_config.gemm1_alpha,
        gemm1_limit=moe_runner_config.gemm1_clamp_limit,
        swiglu_limit=moe_runner_config.swiglu_limit,
    )
    return StandardCombineInput(hidden_states=output)
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 仅新增代码块，不影响现有 CUDA、HIP、CPU 等路径，回归风险低。
  2. XPU 特定内核 fused_experts 的跨线程安全性和数值精度需进一步验证。
  3. 缺少对应的单元测试和模型级测试覆盖。 - 影响：影响范围为 Intel XPU 硬件上使用量化（FP8/MXFP4）的 MoE 模型。性能预期通过 fused_experts 内核获得提升，但无实测数据。对其他平台无影响。 - 风险标记：缺少测试覆盖，XPU 特定数值精度待验证

关联脉络

- PR #22352 :sparkles: [llm][npu][quant] Add W8A8 MXFP8 quantization support for Qwen3 Dense on Ascend NPU: 同样为量化路径添加硬件后端分支，可参考其测试和文档方式。