PR #22654 完整报告

sgl-project/sglang

[XPU] Support apply_router_weight_on_input for Llama4 for fused_experts

合并时间：2026-04-29 10:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22654

执行摘要

- 一句话：XPU fused_experts 支持 router weight on input
- 推荐动作：值得阅读，特别是关注 MoE 架构与后端适配的工程权衡。建议后续追踪 fused_experts 内核内对 apply_router_weight_on_input 的原生支持进展。

功能与动机

Llama4 的 MoE 架构使用了 apply_router_weight_on_input 标志，但 fused_experts 内核并未原生处理该标志。修复此问题后，Llama4 模型可在 XPU 后端上成功推理，且 benchmark 显示 latency 从 3394s 降至 2413s，throughput 从 5.874 token/s 提升至 8.180 token/s。

实现拆解

1. 在 python/sglang/srt/layers/quantization/unquant.py 的 forward_xpu 方法中，从 topk_output 获取 topk_weights 和 topk_ids。
2. 检查 moe_runner_config.apply_router_weight_on_input 是否为 True。
3. 若为 True，则将 x 乘以 topk_weights（按元素乘法，先转换 topk_weights 为 x 的 dtype），然后将 topk_weights 替换为 torch.ones_like，保证 fused_experts 后续行为正确。
4. 后续调用 fused_experts 时使用修改后的 x 和 topk_weights，无需改动 fused_experts 内核。

关键文件：

- python/sglang/srt/layers/quantization/unquant.py（模块 量化层；类别 source；类型 core-logic；符号 forward_xpu）：核心改动文件，在 forward_xpu 中为 fused_experts 添加了 apply_router_weight_on_input 支持。

关键符号：forward_xpu

关键源码片段

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

核心改动文件，在 forward_xpu 中为 fused_experts 添加了 apply_router_weight_on_input 支持。

```
def forward_xpu(self, layer, dispatch_output):
    from sglang.srt.layers.moe.token_dispatcher import StandardCombineInput
```

```
    x = dispatch_output.hidden_states
    topk_output = dispatch_output.topk_output
    moe_runner_config = self.moe_runner_config
    assert moe_runner_config.activation in ["silu", "gelu"]
    backend = self.runner.runner_backend
    if use_intel_xpu_backend():
        from sgl_kernel import fused_experts
        topk_weights, topk_ids, _ = topk_output
        # 如果 apply_router_weight_on_input 为 True（如 Llama4 MoE），
        # 则先将 router weights 应用到输入上，再将 topk_weights 置为全 1，
        # 避免 fused_experts 内重复做加权导致权重被平方
        if moe_runner_config.apply_router_weight_on_input:
            x = x * topk_weights.to(x.dtype)
            topk_weights = torch.ones_like(topk_weights)
        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),
            activation=moe_runner_config.activation,
            gemm1_alpha=moe_runner_config.gemm1_alpha,
            gemm1_limit=moe_runner_config.gemm1_clamp_limit,
        )
        return StandardCombineInput(hidden_states=output)
    else:
        # 其他后端路径不变
        ...
```

评论区精华

Reviewer mingfeima 指出当前实现存在性能问题：topk_weights dtype 转换、乘法、ones_like 均为 outplace 操作，导致额外内存分配。他建议后续可以考虑用 C++ 实现（类似 CPU 的 apply_topk_weights_cpu），或将此标志直接集成到 xpu fused_experts 内核中。PR 作者 rahulvijayaraghavan 表示同意，计划后续扩展 fused_experts 内部支持。最终 mingfeima 审批通过，但标注了 TODO 列表以跟踪内核内的优化。

- apply_router_weight_on_input 的性能实现方式 (performance): 当前方案被接受并合并，同时标注 TODO 以跟踪后续内核级优化。

风险与影响

- 风险：当前实现存在一定的性能风险：额外引入的乘法、类型转换和全 1 张量创建均为 outplace 操作，可能影响推理速度，尤其在 decode 阶段。另外，此实现未经过正式的单元测试验证，回归风险虽低但存在。

- 影响：正面影响：使 Llama4 模型能在 XPU 后端上运行，显著提升推理速度和准确率（从 0.935 到 0.945）。负面影响：微小的性能开销，但对整体优化后的性能提升而言可接受。团队方面，此 PR 为后续将逻辑集成至 fused_experts 内核提供了临时解决方案，也暴露了 xpu 内核需增加此功能的必要性。
- 风险标记：性能开销（额外 outplace 操作），缺少测试覆盖

关联脉络

- 暂无明显关联 PR