PR #24390 完整报告

sgl-project/sglang

[XPU] Enable NVIDIA-Nemotron-3-Nano-30B-A3B-BF16 on Intel XPU backend

合并时间：2026-06-09 09:46

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24390

执行摘要

- 一句话：在 Intel XPU 上启用 Nemotron-3-Nano 混合 Mamba+MoE 模型
- 推荐动作：此 PR 改动简洁，适合学习如何在 SGLang 中为 XPU 适配混合 Mamba+MoE 模型。重点关注跨平台算子绑定（elif is_xpu()）和激活函数白名单扩展的模式。

功能与动机

Before this PR the model crashes at load on XPU: the causal_conv1d import gate only binds on CUDA/NPU, and the MoE path rejects the relu2 activation Nemotron-H uses.

实现拆解

1. Mamba 因果卷积导入适配：在 python/sglang/srt/layers/attention/mamba/mamba.py 中导入 is_xpu，并在 is_npu() 分支后增加 elif is_xpu(): 分支，将 causal_conv1d_fn、causal_conv1d_fn_triton、causal_conv1d_update、causal_conv1d_update_triton 全部绑定至纯 Triton 实现，确保 XPU 无原生 kernel 时仍可正常运行。
2. MoE 激活函数扩展：在 python/sglang/srt/layers/quantization/unquant.py 的 UnquantizedFusedMoEMethod.forward_xpu 方法中，将激活白名单从 ["silu","gelu"] 扩展为 ["silu","gelu","relu2"]，使 sgl-kernel-xpu 的 fused_experts 可以处理 Nemotron-H 使用的平方 ReLU 激活。
3. 测试文件本地化：将端到端测试 test_nvidia_nemotron_3_nano.py 加入 .gitignore，因为该模型需要 tp=4（3B 活跃参数，60GB 权重），无 4-GPU XPU CI runner，测试仅用于本地验证。

关键文件：

- python/sglang/srt/layers/attention/mamba/mamba.py（模块 Mamba 层；类别 source；类型 dependency-wiring）：核心修改：添加 elif is_xpu(): 分支，为 XPU 绑定纯 Triton 实现的 causal_conv1d，是解决模型崩溃的关键。
- python/sglang/srt/layers/quantization/unquant.py（模块 MoE 量化；类别 source；类型 core-logic；符号 forward_xpu）：MoE 路径适配：在 forward_xpu 的激活白名单中增加 relu2，使 fused_experts 支持平方 ReLU。
- .gitignore（模块 忽略规则；类别 other；类型 configuration）：将大模型测试文件 gitignore，避免因硬件不足导致 CI 失败或误提交。

关键符号：forward_xpu

关键源码片段

python/sglang/srt/layers/attention/mamba/mamba.py

核心修改：添加 elif is_xpu(): 分支，为 XPU 绑定纯 Triton 实现的 causal_conv1d，是解决模型崩溃的关键。

```
from sglang.srt.utils import (
    is_cpu,
    is_cuda,
    is_npu,
    is_xpu, # 新增导入用于 XPU 平台判断
    set_weight_attrs,
)

if is_cuda():
    from sglang.srt.layers.attention.mamba.causal_conv1d import (
        causal_conv1d_fn,
        causal_conv1d_update,
    )
    from sglang.srt.layers.attention.mamba.causal_conv1d_triton import (
        causal_conv1d_fn as causal_conv1d_fn_triton,
        causal_conv1d_update as causal_conv1d_update_triton,
    )
elif is_npu():
    from sgl_kernel_npu.mamba.causal_conv1d import (
        causal_conv1d_fn_npu as causal_conv1d_fn,
        causal_conv1d_update_npu as causal_conv1d_update,
    )
elif is_xpu():
    # XPU 尚无原生 causal_conv1d kernel，因此将纯 Triton 实现绑定到
    # `causal_conv1d_fn` / `causal_conv1d_fn_triton` / `causal_conv1d_update`
    # / `causal_conv1d_update_triton` 四个入口点，确保所有代码路径都可调用。
    from sglang.srt.layers.attention.mamba.causal_conv1d_triton import (
        causal_conv1d_fn as causal_conv1d_fn,
        causal_conv1d_fn as causal_conv1d_fn_triton,
        causal_conv1d_update as causal_conv1d_update,
        causal_conv1d_update as causal_conv1d_update_triton,
    )
```

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

MoE 路径适配：在 forward_xpu 的激活白名单中增加 relu2，使 fused_experts 支持平方 ReLU。

```
def forward_xpu(self, layer, dispatch_output):
    # ... 前置计算 ...
    moe_runner_config = self.moe_runner_config
    assert moe_runner_config.activation in [
```

```
        "silu",
        "gelu",
        "relu2", # Nemotron-H 使用平方 ReLU，此前不在白名单中导致崩溃
    ], f"activation = {moe_runner_config.activation} is not supported."

    backend = self.runner.runner_backend
    if use_intel_xpu_backend():
        # sgl-kernel-xpu 路径：fused_experts 已支持 relu2
        from sgl_kernel import fused_experts
        # ... 调用 fused_experts ...
    else:
        # Triton 路径仅允许 silu；relu2 不会走到这里
        assert backend.is_triton()
        assert (
            moe_runner_config.activation == "silu"
        ), f"activation = {moe_runner_config.activation} is not supported for Triton PATH, please
        set ENV SGLANG_USE_SGL_XPU=1."
```

评论区精华

核心讨论包括：

- mingfeima质疑 mamba.py 的 XPU 分支是否必要，Xia-Weiwen解释 Qwen3.5 走不同路径（GDN 后端），而 Nemotron 确实需要此分支。
- gemini-code-assist[bot]建议将重复的 _device_context 辅助函数提取到共享模块，但该文件最终未包含在合并中。
- gemini-code-assist[bot]指出 forward_xpu 中 relu2 在断言处是死代码，作者在后续提交中修复了 Triton 路径的断言。
- 测试脚本 device 参数设置 (correctness): 该测试文件最终被移出版本控制并加入 .gitignore，因此未修改。
- mamba.py XPU 分支必要性 (design): 确认该分支是必需的，因为 Nemotron 模型确实走 mamba 路径。
- relu2 断言死代码 (correctness): 作者在后续提交中修复了 Triton 路径的断言，仅保留 silu，消除了死代码。
- 重复的 _device_context 辅助函数 (design): 未采纳，相关文件最终未包含在合并中。

风险与影响

- 风险：
  1. 性能风险：XPU 使用纯 Triton 实现 causal_conv1d，没有原生 kernel，可能带来性能开销。
  2. 兼容性风险：relu2 仅在 XPU 的 fused_experts 路径中支持，若未来其他后端需要类似激活，需单独适配。
  3. 回归风险：端到端测试未纳入 CI，模型行为退化不易被自动化检测。

4. 维护风险：增加的 XPU 条件分支需与上游 Triton 版本保持兼容。 - 影响：用户影响：XPU 用户现在可以运行 Nemotron-3-Nano 模型进行推理，获得与 B200 相当的 GSM8K 准确率。系统影响：改动仅增加条件分支和激活白名单，不改变已有路径行为。团队影响：为 XPU 引入特定分支维护点，需关注上游 causal_conv1d_triton 代码的稳定性。

- 风险标记：缺少 CI 测试覆盖，XPU 依赖纯 Triton 实现，relu2 仅 XPU 后端支持

关联脉络

- 暂无明显关联 PR