PR #21698 完整报告

sgl-project/sglang

[npu]fix: qwen3-next w8a8 precision bugs

合并时间：2026-04-27 18:14

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21698

执行摘要

- 一句话：修复 NPU 上 Qwen3-Next W8A8 精度问题
- 推荐动作：此 PR 属于 必要且精准的 Bug 修复，建议尽快合并。核心价值在于揭示了 W8A8 量化模型中 _override_weight_loader 的潜在陷阱——当涉及融合投影和多个量化参数时，必须迭代所有相关参数。设计上值得关注的是其遍历参数列表的模式，可推广到其他类似场景作为最佳实践。建议后续添加针对 W8A8 量化参数 loader 覆写的单元测试，以防止回归。

功能与动机

PR #19321 的变更导致 Qwen3-Next 在 NPU 上运行时 W8A8 模型失败，原因是 in_proj_qkvz 的量化参数加载不正确。PR body 明确指出："The change at https://github.com/sgl-project/sglang/pull/19321 caused qwen3-next to fail in running the w8a8 model on NPU due to incorrect loading of the in_proj_qkvz quantization parameters."

实现拆解

变更仅涉及一个文件 python/sglang/srt/models/qwen3_next.py，包含两个关键修复：

1. 扩展 _override_weight_loader 方法（+11/-5 行）：原方法仅对 module.weight 应用 new_loader。修改后，方法改为遍历 weight、weight_scale_inv、weight_scale、input_scale、weight_offset 五个属性，对每个非 None 的参数重写其 weight_loader。这使得所有 W8A8 量化相关的参数（权重、缩放因子、偏移量）都能正确继承自定义的 loader 逻辑，确保量化参数在 TP 切片、融合权重加载等场景下被正确处理。
2. NPU 环境下算子替换（+8/-0 行）：在文件顶部新增 if _is_npu: 条件块，从 sgl_kernel.npu.fla.utils 导入 fused_qkvzba_split_reshape_cat_npu 并覆盖全局的 fused_qkvzba_split_reshape_cat 引用。这样做的原因是原有 Triton 算子在 prefill 阶段 grid 尺寸可能超过 65535 的限制，而 NPU 专用实现（sgl_kernel_npu 中的版本）可以避免此问题。

无测试、配置或部署相关的配套改动。

关键文件：

- python/sglang/srt/models/qwen3_next.py（模块 模型加载；类别 source；类型 data-contract；符号 _override_weight_loader, _make_packed_weight_loader）：唯一的

变更文件，包含两个核心修复：扩展量化参数 weight_loader 覆盖和 NPU 下算子替换。

关键符号：_override_weight_loader, _make_packed_weight_loader

关键源码片段

python/sglang/srt/models/qwen3_next.py

唯一的变更文件，包含两个核心修复：扩展量化参数 weight_loader 覆盖和 NPU 下算子替换。

```
# python/sglang/srt/models/qwen3_next.py

# QwenGatedDeltaNet 的静态方法
@staticmethod
def _override_weight_loader(module, new_loader):
    """Override weight_loader on a module's weight parameter.

    ModelWeightParameter exposes weight_loader as a read-only property
    backed by _weight_loader, while plain parameters store it as a
    regular attribute.  This helper handles both cases."""
    # 修复前：只对 module.weight 设置新 loader
    # 修复后：遍历所有量化相关参数，对每个非 None 的参数设置 loader
    for attr_name in (
        "weight",
        "weight_scale_inv",
        "weight_scale",
        "input_scale",
        "weight_offset",
    ):
        param = getattr(module, attr_name, None)
        if param is None:
            continue
        if hasattr(param, "_weight_loader"):
            param._weight_loader = new_loader
        else:
            param.weight_loader = new_loader

# 文件顶部新增的条件导入（用于 NPU 环境）
# 在 class Qwen3GatedDeltaNet 定义之前
if _is_npu:
    from sgl_kernel_npu.fla.utils import (
        fused_qkvzba_split_reshape_cat as fused_qkvzba_split_reshape_cat_npu,
    )
    # 将全局的 fused 函数替换为 NPU 专用版本
    # 目的：避免 Triton 算子 grid 尺寸超过 65535（prefill 阶段）
    fused_qkvzba_split_reshape_cat = fused_qkvzba_split_reshape_cat_npu
```

评论区精华

该 PR 的 review 评论较少。gemini-code-assist[bot] 总结了变更内容并表示无额外反馈。sglang-npu-bot 直接批准了 PR。从技术角度看，_override_weight_loader 的改动是关键的修

正：原实现只处理weight参数，但W8A8量化在融合投影层（MergedColumnParallelLinear）中，量化参数如 weight_scale 和 weight_offset 也需要相同的自定义 loader 来正确处理 TP 分片，否则会导致错误的分片甚至精度问题。这点在实现中并无深入讨论，但属于合理的推断。

- 变更确认与自动化审核 (other)：双方无异议，PR 被批准。

风险与影响

- 风险：
  1. 回归风险：_override_weight_loader 的改动是迭代所有量化参数，这不会影响原有逻辑（原有逻辑被保留在循环中），但若某些子模块不存在这些属性，getattr 返回 None 后跳过，行为安全；不过对于未来新增的量化参数，可能需要同步更新该循环。
  2. 算子替换风险：NPU 环境下 fused_qkvzba_split_reshape_cat 被替换为 sgl_kernel_npu 版本，但这仅限于 _is_npu=True 时，对 CUDA 等其他后端无影响。但该替换未经严格等价测试，可能存在细微语义差异，建议通过 NPU 上的预填 / 解码回归测试覆盖。
  3. 测试覆盖不足：PR 未包含任何单元测试或集成测试来验证修复的正确性，仅依赖于后续 CI 中的模型精度测试。 - 影响：影响范围：仅限于 NPU 环境下使用 W8A8 量化加载的 Qwen3-Next 模型。对于 CUDA/CPU 后端无影响。影响程度：严重程度较高，因为该 Bug 导致 NPU 上 Qwen3-Next W8A8 模型完全无法运行，修复后可恢复其功能。用户影响：NPU 用户在使用 W8A8 量化的 Qwen3-Next 模型时，之前会遇到加载失败或精度问题，修复后模型可正常使用。
- 风险标记：缺少测试覆盖，NPU 专用分支

关联脉络

- PR #19321 [相关 PR] 触发本次 Bug 的原始变更：PR #19321 的修改导致 W8A8 量化参数的 loader 未被正确覆写，是当前 PR 的直接修复对象。
- PR #20918 [NPU] Support MTP for Qwen3.5: 与当前 PR 同属 NPU 平台上的 Qwen3 系列模型适配工作。