PR #22352 完整报告

sgl-project/sglang

:sparkles: [llm][npu][quant] Add W8A8 MXFP8 quantization support for Qwen3 Dense on Ascend NPU

合并时间：2026-06-16 14:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22352

执行摘要

- 一句话：为 Ascend NPU Qwen3 Dense 模型添加 W8A8 MXFP8 量化支持
- 推荐动作：值得精读，尤其 process_weights_after_loading 中对 .contiguous() 的差异化处理是 NPU 量化的关键设计决策，显示了平台硬件特性对软件实现的约束。

功能与动机

参考 PR body："closes part of the NPU quantization gap tracked in issue #21584"。目标是支持 Ascend NPU A5 系列硬件的 MXFP8 量化，以降低内存带宽和计算开销，同时保持接近无损精度。

实现拆解

1. NPU 旁路与分发：在 fp8.py 的 Fp8Config.get_min_capability() 中增加 NPU 返回 0 的旁路，避免 CUDA capability 检查；在 get_quant_method() 中增加 NPU+use_mxfp8 分支，分发到 NPUMXFP8LinearMethod。
2. 在线量化线性方法：在 linear_method_npu.py 新增 NPUMXFP8LinearMethod 类。create_weights 以原始 FP16/BF16 dtype 预留权重；process_weights_after_loading 通过 npu_dynamic_mx_quant 在线量化为 MXFP8 并预转置为 [in, out] 布局（不调用 .contiguous() 以保留分析视图）；apply 对激活逐 token 量化后调用 npu_quant_matmul（group_sizes=[1,1,32]）。
3. 离线量化方案：在 modelslim_mxfp8.py 新增 ModelSlimMXFP8Scheme 类，处理 msmodelslim 预量化权重（float8_e4m3fn + uint8 scale）。create_weights 创建对应 dtype 的参数；process_weights_after_loading 转置但不调用 .contiguous() 以保持 block-scale 映射；apply_weights 类似在线路径进行激活量化和矩阵乘法。
4. 注册与导入调整：在 modelslim/modelslim.py 的 get_linear_scheme 表中注册 ("W8A8_MXFP8", ModelSlimMXFP8Scheme)；在 schemes/__init__.py 中先导入基类再导入新 scheme 以避免循环依赖。
5. 导入修复：在 rotary_embedding/base.py 中将 fused_rope_qk_mqa 导入包裹在 try/except ImportError 中，并将符号置为 None，避免缺失 kernel 导致整个模块导入失败进而使模型静默回退。

6. 文档：更新 quantization.mdx 添加 mxfp8 行；更新 ascend_npu_quantization.mdx 补充 LLM dense 使用示例。

关键文件：

- python/sglang/srt/layers/quantization/modelslim/schemes/modelslim_mxfp8.py（模块 量化方案；类别 source；类型 data-contract；符号 ModelSlimMXFP8Scheme, init, create_weights, process_weights_after_loading）：新增离线量化方案，实现预量化权重的加载、转置和 MXFP8 推理，是离线路径的核心。
- python/sglang/srt/hardware_backend/npu/quantization/linear_method_npu.py（模块 量化实现；类别 source；类型 core-logic；符号 NPUMXFP8LinearMethod, create_weights, process_weights_after_loading, apply）：新增 NPUMXFP8LinearMethod 在线量化类，实现 FP16/BF16 在线量化为 MXFP8 并执行量化矩阵乘法，是量化推理的关键执行路径。
- python/sglang/srt/layers/rotary_embedding/base.py（模块 位置编码；类别 source；类型 dependency-wiring）：修复缺失 fused_rope_qk_mqa kernel 时模块导入失败的问题，通过 try/except 容忍缺失，并在 forward_npu 中检查 None 后回退。
- python/sglang/srt/layers/quantization/fp8.py（模块 量化框架；类别 source；类型 dependency-wiring）：添加 NPU 旁路（get_min_capability 返回 0）和分发逻辑（get_quant_method 返回 NPUMXFP8LinearMethod），使 --quantization mxfp8 能在 NPU 上触发正确路径。
- python/sglang/srt/layers/quantization/modelslim/schemes/__init__.py（模块 量化方案；类别 source；类型 data-contract）：导出 ModelSlimMXFP8Scheme 并调整导入顺序避免循环依赖，是离线量化方案的注册入口。
- python/sglang/srt/layers/quantization/modelslim/modelslim.py（模块 量化注册；类别 source；类型 data-contract）：注册 ("W8A8_MXFP8", ModelSlimMXFP8Scheme) 到 get_linear_scheme 表，使离线量化方案可在配置中通过名称查找。
- docs_new/docs/hardware-platforms/ascend-npus/ascend_npu_quantization.mdx（模块 文档；类别 other；类型 core-logic）：更新 Ascend NPU 量化文档，添加 LLM dense MXFP8 在线 / 离线使用示例。
- docs_new/docs/advanced_features/quantization.mdx（模块 文档；类别 other；类型 core-logic）：在跨平台量化支持表中增加 mxfp8 行，标注 Ascend A5 支持。

关键符号：NPUMXFP8LinearMethod.create_weights, NPUMXFP8LinearMethod.process_weights_after_loading, NPUMXFP8LinearMethod.apply, ModelSlimMXFP8Scheme.create_weights, ModelSlimMXFP8Scheme.process_weights_after_loading, ModelSlimMXFP8Scheme.apply_weights, Fp8Config.get_min_capability, Fp8Config.get_quant_method, ModelSlimConfig.get_linear_scheme

关键源码片段

python/sglang/srt/layers/rotary_embedding/base.py

修复缺失 fused_rope_qk_mqa kernel 时模块导入失败的问题，通过 try/except 容忍缺失，并在 forward_npu 中检查 None 后回退。

```
if _is_npu:
    import torch_npu

    # `fused_rope_qk_mqa` is an optional fast-path kernel shipped with
    # `sgl_kernel_npu`. Older NPU CANN / sgl_kernel_npu builds may not
    # include it. If we let the ImportError propagate, importing this
    # module fails, which in turn causes `ModelRegistry` to silently skip
    # every model that depends on it (and fall back to HF Transformers
    # without quantisation awareness — see PR #22352). We tolerate the
    # missing kernel so model loading still works; call sites must check
    # for `None` and use the generic rope path. A warning is emitted so
    # the missing kernel is visible in logs instead of being silently
    # swallowed.
    try:
        from sgl_kernel_npu.norm.fused_rope_qk_mqa import fused_rope_qk_mqa
    except ImportError:
        fused_rope_qk_mqa = None
        logger.warning(
            "sgl_kernel_npu.norm.fused_rope_qk_mqa is unavailable; "
            "falling back to the generic rope implementation.  Upgrade "
            "sgl_kernel_npu to enable the fused kernel."
        )

# ... inside forward_npu:
        if (
            fused_rope_qk_mqa is not None
            and query.shape[0] * query.shape[1] < 65535
        ):
            return fused_rope_qk_mqa(
                query,
                key,
                cos_sin,
                self.rotary_dim,
                self.is_neox_style,
            )
        else:
            return self.forward_native(positions, query, key, offsets)
```

评论区精华

- 预转置与 block-scale 映射：gemini-code-assist 建议离线量化权重预转置时不调用 .contiguous()，作者确认已实现并解释原因。
- ImportError 容忍：ping1jing2 问为什么跳过 fused_rope_qk_mqa 导入错误，作者说明缺失 kernel 会导致模块导入失败进而使 ModelRegistry 静默回退，已改为 try/except+warning。

- weight dtype 告警：ping1jing2 建议对非预期 weight dtype 添加 warning，作者接受并添加。
- 方案委托 kernel：TamirBaydasov 建议 ModelSlimMXFP8Scheme 应委托处理逻辑给 NPUMXFP8LinearMethod，作者同意但计划在后续 PR 中重构。
  - 预转置权重和 scale 时不调用 .contiguous() 以保持 block-scale 映射 (correctness): 作者确认已按建议实现，在 modelslim_mxfp8.py 中使用 .data.transpose() 而不调用 .contiguous()。
  - 缺失 fused_rope_qk_mqa 时跳过 ImportError 的原因 (correctness): 作者使用 try/except 并将 fused_rope_qk_mqa 置为 None，同时发出 warning。
  - 在线量化路径中应添加 weight dtype 告警 (design): 作者接受并在 6f0e711 中添加了 logger.warning。
  - ModelSlimMXFP8Scheme 应委托 kernel 给 NPUMXFP8LinearMethod (design): 作者同意但计划在后续 PR 中重构，以免影响当前合并。

风险与影响

- 风险：
  1. NPU 条件导入：虽然使用 current_platform.is_npu() 保护，但若平台检测出错可能导致 CUDA/CPU 上错误导入 torch_npu 而崩溃。
  2. 布局差异风险：在线与离线路径对 .contiguous() 的处理不同（在线调用、离线不调用），若后续修改未注意此差异可能引入难以调试的数值错误。
  3. rotary_embedding 修改影响面：该修改影响所有 NPU 模型加载，若 try/except 未覆盖其他缺失 kernel 场景可能导致静默回退降低推理精度。
  4. 缺少测试覆盖：没有直接添加单元测试或端到端测试，回归风险依赖 NPU CI。 - 影响：用户：Ascend NPU A5 用户可通过 --quantization mxfp8 或 msmodelslim 预量化权重获得 MXFP8 加速；非 NPU 用户无影响。系统：新引入 NPUMXFP8LinearMethod 和 ModelSlimMXFP8Scheme，代码量适中，与原量化框架集成良好。团队：后续需关注已识别的重构项（委托 kernel、统一 torch.ops.npu），并补充测试。
- 风险标记：核心路径变更 (rotary_embedding)，缺少直接测试覆盖，NPU 专用代码条件保护需验证

关联脉络

- PR #20922 [Diffusion] MXFP8 Quantization Support on Ascend NPU: 前提 PR，提供了 NPU 量化基础设施（MXFP8Config、NPUMXFP8DiffusionLinearMethod），本 PR 在其基础上构建 LLM 支持。
- PR #22338 [Diffusion] MXFP4 Quantization Support on Ascend NPU: 前提 PR，扩展量化框架支持 Diffusion MXFP4，本 PR 与之共享 NPU 量化工具函数。