PR #23654 完整报告

sgl-project/sglang

[MUSA][20/N] Support qwen series models

合并时间：2026-05-01 02:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23654

执行摘要

- 一句话：支持 Qwen 系列模型在 MUSA 后端运行
- 推荐动作：建议重点审查新增 topk kernel 的性能与正确性（特别是 autotune 配置在实际生产中的适用性），并在 CI 中为 MUSA 增加基本回归测试。MoE 路由分支的维护者应关注 mate 库的更新同步。整体 PR 设计合理，适合合并。

功能与动机

作为 MUSA 后端支持系列（19/N）的一环，目标是在 MUSA 平台上启用 Qwen 系列模型推理。需要补齐 MUSA 专用的 MoE fused gate 与 top-k kernel，并解决 vision attention、FP8 量化以及多平台 dispatch 中的兼容性问题。

实现拆解

1. 新增 MUSA TopK Kernel(python/sglang/srt/hardware_backend/musa/kernels/topk.py，新增 300 行)：用 Triton 实现了 topk_softmax_triton_kernel 和 topk_sigmoid_triton_kernel，支持 correction_bias、renormalize、moe_softcapping。BLOCK_K 改为 next_power_of_two(K) 修复硬编码 16 问题；去除 float16 中间转换以提升精度；消除重复 softmax 计算。
2. MoE 路由集成(python/sglang/srt/layers/moe/topk.py)：将 MUSA 从 sgl_kernel.moe_fused_gate 联合导入中分离，单独从 mate 库导入 moe_fused_gate。在 biased_grouped_topk_gpu 中添加 elif _is_musa 分支，条件满足时调用 mate.moe_fused_gate，否则回退 PyTorch 原生路径。topk_softmax/topk_sigmoid 导入切换为 MUSA 后端内核。
3. FlashAttention 多步后端(python/sglang/srt/hardware_backend/musa/attention/flashattention_backend.py)：新增 MusaFlashAttentionMultiStepBackend，继承自 CUDA 版 FlashAttentionMultiStepBackend，内部循环创建 MusaFlashAttentionBackend。移除 init_forward_metadata 中重复的 extend_with_prefix 判断。
4. Vision Attention 与 Speculative Decoding 适配(vision.py, draft_utils.py, 多个 speculative 文件)：vision.py 添加 _is_musa 支持并在 _determine_attention_backend 中根据算力选择 FA3 或 Triton；draft_utils.py 在创建 FA 后端时根据 MUSA 加载对应类；其他 speculative 模块条件化 sgl_kernel 导入。

5. FP8 量化路径修正(fp8_kernel.py, fp8_utils.py)：移除 MUSA 专有 .contiguous() 调用；deepgemm_w8a8_block_fp8_linear_with_fallback 中 MUSA 分支跳过 column_major_scales/scale_tma_aligned/scale_ue8m0，仅使用连续 scale。

测试配套：PR 描述附带了 Qwen3.5-27B-FP8 和 DeepSeek 模型的实际推理日志与准确率结果，但未在 test/ 目录下新增自动化回归用例。

关键文件：

- python/sglang/srt/hardware_backend/musa/kernels/topk.py（模块 MUSA 内核；类别 source；类型 core-logic；符号 tanh, topk_softmax_triton_kernel, topk_softmax, topk_sigmoid_triton_kernel）：新增 300 行的 Triton kernel，实现 MoE 路由所需的 topk_softmax 和 topk_sigmoid，是 MUSA 后端最核心的计算补充。
- python/sglang/srt/hardware_backend/musa/attention/flashattention_backend.py（模块 注意力层；类别 source；类型 core-logic；符号 init_forward_metadata, MusaFlashAttentionMultiStepBackend, init）：新增 MusaFlashAttentionMultiStepBackend，并调整 init_forward_metadata 逻辑，是 speculative decoding 在 MUSA 路径的关键适配。
- python/sglang/srt/layers/moe/topk.py（模块 MoE 路由；类别 source；类型 dependency-wiring；符号 biased_grouped_topk_gpu, moe_fused_gate）：MoE 路由的核心调度文件，将 MUSA 与 CUDA 路径分离，新增 mate.moe_fused_gate 调用和 topk_softmax/sigmoid 的 MUSA 导入。
- python/sglang/srt/speculative/draft_utils.py（模块 推测解码；类别 source；类型 dependency-wiring；符号 is_musa, _create_fa_decode_backend, _create_fa_prefill_backend）：MUSA 分支注入：在创建 FA 后端时根据 is_musa() 加载 MusaFlashAttentionMultiStepBackend 或 MusaFlashAttentionBackend。
- python/sglang/srt/layers/attention/vision.py（模块 视觉模块；类别 source；类型 dependency-wiring；符号 VisionFlash3Attention, _determine_attention_backend）：为 VisionFlash3Attention 添加 MUSA 支持，并修改后端选择逻辑以适配 MUSA 设备。
- python/sglang/srt/layers/quantization/fp8_utils.py（模块 量化层；类别 source；类型 core-logic；符号 deepgemm_w8a8_block_fp8_linear_with_fallback）：修改 deepgemm_w8a8_block_fp8_linear_with_fallback，MUSA 分支跳过 MUSA 不支持的量化参数。
- python/sglang/srt/layers/quantization/fp8_kernel.py（模块 量化层；类别 source；类型 core-logic；符号 w8a8_block_fp8_matmul_deepgemm）：移除 MUSA 专有的 .contiguous() 调用，避免非连续 tensor 错误。

关键符号：topk_softmax, topk_sigmoid, topk_softmax_triton_kernel, topk_sigmoid_triton_kernel, MusaFlashAttentionMultiStepBackend, biased_grouped_topk_gpu, deepgemm_w8a8_block_fp8_linear_with_fallback, _create_fa_decode_backend, _create_fa_prefill_backend, VisionFlash3Attention.init, _determine_attention_backend

关键源码片段

python/sglang/srt/layers/moe/topk.py

MoE 路由的核心调度文件，将 MUSA 与 CUDA 路径分离，新增 mate.moe_fused_gate 调用和 topk_softmax/sigmoid 的 MUSA 导入。

```
# 导入部分变更
if _is_musa:
    try:
        from mate import moe_fused_gate
    except ImportError as e:
        raise ImportError("mate is required for the biased grouped topk.")

    from sglang.srt.hardware_backend.musa.kernels.topk import topk_sigmoid, topk_softmax

# biased_grouped_topk_gpu 中的 MUSA 分支（约 887 行附近）
elif _is_musa and (
    gating_output.shape[1] // num_expert_group <= 32
    or (num_expert_group == 1 and gating_output.shape[1] in {160, 256, 384})
):
    # 调用 MUSA 专用 fused gate kernel
    topk_weights, topk_ids = moe_fused_gate(
        gating_output.to(dtype=torch.float32),
        correction_bias,
        num_expert_group,
        topk_group,
        topk,
        num_fused_shared_experts,
        routed_scaling_factor if routed_scaling_factor is not None else 1.0,
        True, # renormalize 硬编码为 True（kernel 内部行为）
        apply_routed_scaling_factor_on_output,
    )
else:
    # 回退到通用路径（如 PyTorch 原生或其他 GPU 后端）
    ...
```

评论区精华

- Kernel 硬编码 BLOCK_K：gemini-code-assist[bot] 指出 topk kernel 中 BLOCK_K 硬编码为 16，当 topk>16 时越界。作者修复为动态 next_power_of_two(K)。
- MoE 路由条件矛盾：允许专家数 160/384 的列表但又要求 correction_bias 长度为 2 的幂，导致这些配置被跳过。作者已修复。
- FP8 回归风险：yeahdongcn 要求确认移除 MUSA 专用分支后的 DeepSeek 模型表现。作者提供了 GSM8K 准确率数据（DeepSeek-Coder-V2-Lite 82%，DeepSeek-V2-Lite-FP8 67%），popsiclexu 确认非连续 tensor 错误已通过参数调整解决。
  - topk kernel 中 BLOCK_K 硬编码为 16 (correctness): 作者已修复（fixed）。
  - MoE 路由中 power-of-two 条件与专家数 160/384 矛盾 (correctness): 作者回复已修复（fixed）。

- FP8 kernel 中 MUSA 工作区移除的回归风险 (performance): 通过测试验证，结论可接受。

风险与影响

- 风险:
  - 新 Kernel 正确性：新增 Triton kernel 虽通过 autotune 但未在 test/ 下集成自动化测试，边界条件（如 topk 变化、极值专家数）缺乏覆盖。
  - FP8 量化回归：修改了 fp8_kernel.py 和 fp8_utils.py 中 MUSA 分支，可能影响 DeepSeek 系列模型精度；review 提供了部分验证但非系统性。
  - Speculative Decoding 兼容性：MUSA 上的 spec 解码路径仅通过条件分支切换后端，未针对 MUSA 特性优化，可能性能不佳或存在隐藏 bug。
  - 跨平台耦合：MoE 路由中新增的 MUSA 条件分支与现有的 CUDA/HIP/XPU 逻辑并列，增加了维护复杂度。
- 影响:
  - 用户：MUSA 平台用户可运行 Qwen 系列模型（包括 Qwen2、Qwen3.5 等），FP8 和 speculative decoding 场景可正常工作。
  - 系统：改动集中在硬件后端目录和条件分支，CUDA/AMD 等主路径无运行时影响（编译期条件化）。
  - 团队：需持续维护 MUSA 专用 kernel，并保持与 CUDA 版功能的同步演进。
  - 风险标记：新硬件后端 Kernel 正确性，FP8 量化路径回归，缺少专项测试覆盖，Speculative Decoding 兼容性

关联脉络

- 暂无明显关联 PR