PR #23280 完整报告

sgl-project/sglang

[XPU] Enable Gemma 4 E2B / E4B / 31B/ 26B-A4B on Intel XPU

合并时间：2026-06-05 10:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23280

执行摘要

- 一句话：在 Intel XPU 上启用 Gemma 4 系列模型
- 推荐动作：建议精读：xpu_backend.py 中的 SWA 页表翻译设计和 gemma4_fused_ops.py 中的路由融合 kernel，这两个是 XPU 后端适配混合注意力模型的关键创新。整体架构清晰，改动自包含，值得参考。值得关注的决策：将 fused QKV RMSNorm 断言放宽为 is_cuda or is_xpu 并依赖 Triton JIT 的设备无关性，以及通过 fuse_scale 将 scale 折叠进 norm.weight 减少 kernel launch。

功能与动机

启用 Google Gemma 4 系列模型在 Intel XPU 上的推理。Gemma 4 是混合注意力模型，交错使用滑动窗口注意力（SWA，head_dim=256）和全注意力（head_dim=512），在 XPU 上需要内核级适配才能正确运行。PR body 明确指出 Enable google/gemma-4-E2B-it、google/gemma-4-E4B-it、google/gemma-4-31B-it 和 google/gemma-4-26B-A4B-it on Intel XPU with the intel_xpu attention backend。

实现拆解

1. 修复 RMSNorm 对 >2D 输入的支持：在 layernorm.py 中为 RMSNorm.forward_xpu 移除 3D→2D 的 reshape（sgl-kernel-xpu 原生处理 stride），并为 Gemma4RMSNorm.forward_xpu 提供自包含实现，不再委托 forward_cuda。
2. 实现混合 SWA KV 池的页表翻译：在 xpu_backend.py 中检测 SWAKVPool，在 init_forward_metadata 和 forward_extend/forward_decode 中通过 translate_loc_from_full_to_swa 将全池页表索引转换为 SWA 池索引，并处理 page_size > 1 时的 strided 格式。同时支持跨层 KV 共享（k=None, v=None 路径）。
3. 放宽 fused Triton kernel 的设备断言：在 gemma4_fused_ops.py 中将 gemma_qkv_rmsnorm 的 is_cuda 断言改为 is_cuda or is_xpu，并在 gemma4_causal.py 中允许 XPU 进入 gemma_rmsnorm_residual_scalar 和 gemma_dual_rmsnorm_residual_scalar 的 fused 路径。
4. 新增融合路由后处理 kernel：在 gemma4_fused_ops.py 中添加 _gemma_routing_post_topk_kernel 和 gemma_routing_post_topk，将 softmax、per_expert_scale 乘法和类型转换融合为单个 Triton kernel，并在 gemma4_causal.py 的 routing_function 中根据 is_xpu 或 is_cuda 调用该融合路径。

5. 更新服务器配置和测试：在 server_args.py 中将 intel_xpu 加入 Gemma 4 接受的 attention backend 列表。新增 test/registered/xpu/test_gemma_4_e2b.py，包含简单 Q&A、SWA 长上下文和 SWA 3K tokens 三个测试用例，并注册到 CI。

关键文件：

- test/registered/xpu/test_gemma_4_e2b.py（模块 测试套件；类别 test；类型 test-coverage；符号 _simple_text_messages, _empty_xpu_cache, TestGemma4E2BXPU, setUpClass）：新增 E2B 烟囱测试，验证模型在 XPU 上的基本功能、SWA 长上下文和 3K tokens，是 CI 门禁的核心测试。
- python/sglang/srt/layers/gemma4_fused_ops.py（模块 路由融合；类别 source；类型 core-logic；符号 _gemma_routing_post_topk_kernel, gemma_routing_post_topk）：核心变更：放宽 gemma_qkv_rmsnorm 的 is_cuda 断言以支持 XPU；新增融合路由后处理 Triton kernel，减少 kernel launch 数量。
- python/sglang/srt/layers/attention/xpu_backend.py（模块 注意力层；类别 source；类型 dependency-wiring；符号 use_sliding_window_kv_pool, init_forward_metadata, forward_extend, forward_decode）：核心适配：检测 SWAKVPool 并实现页表翻译；支持跨层 KV 共享；允许 XPU 进入 fused QKV RMSNorm 路径。
- python/sglang/srt/models/gemma4_causal.py（模块 模型定义；类别 source；类型 data-contract；符号 Gemma4Router, fuse_scale, routing_function, forward）：模型定义层调整：引入融合路由 kernel 的调用；修改 fuse_scale 以折叠 scale 到 norm.weight；允许 XPU 进入 A1/A2 融合归一化路径。
- python/sglang/srt/layers/layernorm.py（模块 归一化层；类别 source；类型 core-logic；符号 forward_xpu）：归一化层适配：提供自包含的 Gemma4RMSNorm.forward_xpu，移除 3D→2D reshape，避免依赖 forward_cuda。
- python/sglang/srt/server_args.py（模块 服务器配置；类别 source；类型 core-logic；符号 _handle_model_specific_adjustments）：服务器配置调整：将 intel_xpu 加入 Gemma 4 接受的 attention backend 白名单。

关键符号：forward_xpu (RMSNorm), forward_xpu (Gemma4RMSNorm), gemma_qkv_rmsnorm, gemma_routing_post_topk, fuse_scale (Gemma4Router), routing_function, init_forward_metadata (XPUAttentionBackend), forward_extend (XPUAttentionBackend), forward_decode (XPUAttentionBackend)

关键源码片段

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

核心适配：检测 SWAKVPool 并实现页表翻译；支持跨层 KV 共享；允许 XPU 进入 fused QKV RMSNorm 路径。

```
# 路径：python/sglang/srt/layers/attention/xpu_backend.py

# 在 __init__ 中检测是否使用 SWAKVPool
class XPUAttentionBackend(AttentionBackend):
    def __init__(self, model_runner, ...):
        # ...
```

```
        self.is_hybrid_swa = model_runner.is_hybrid_swa
        # [!] 新增：检测 token_to_kv_pool 是否为 SWAKVPool 且包含 SWA 层
        self.use_sliding_window_kv_pool = (
            isinstance(model_runner.token_to_kv_pool, SWAKVPool)
            and model_runner.token_to_kv_pool.swa_layer_nums > 0
        )
        if self.use_sliding_window_kv_pool:
            self.token_to_kv_pool = model_runner.token_to_kv_pool
        # ...

    def init_forward_metadata(self, forward_batch: ForwardBatch):
        # ...
        # 准备常规 page_table
        metadata.page_table = self.req_to_token_pool.req_to_token[
            forward_batch.req_pool_indices, :metadata.max_seq_len_k
        ]

        # [!] 新增：翻译全池索引到 SWA 池索引（混合模型使用）
        if self.use_sliding_window_kv_pool:
            metadata.swa_page_table = (
                self.token_to_kv_pool.translate_loc_from_full_to_swa(
                    metadata.page_table
                ).to(torch.int32)
            )
        # ...
        # 处理 page_size > 1 时的 strided 格式
        if self.page_size > 1:
            if self.use_sliding_window_kv_pool and metadata.swa_page_table is not None:
                metadata.swa_page_table = (
                    metadata.swa_page_table[:, self.strided_indices] // self.page_size
                )
            metadata.page_table = (
                metadata.page_table[:, self.strided_indices] // self.page_size
            )

    def forward_extend(self, q, k, v, ...):
        # [!] 新增：跨层 KV 共享 (k=None, v=None) 的支持
        if k is None and v is None:
            # Cross-layer KV sharing: 跳过 store_cache 等操作
            # ...
```

评论区精华

Review 中主要讨论集中在以下几个主题：

- RMSNorm reshape 争议：airMeng 询问为什么 cuda 不需要这个 reshape，jmunetong 指出 cuda 也有类似 reshape。mingfeima 建议修改 fused_add_rmsnorm 以原生处理 2D 和 3D 输入，避免 view 和内存拷贝。最终实现中，forward_xpu 不再包含显式 reshape，依赖 sgl-kernel-xpu 的 stride 处理。

- dispatch 歧义：mingfeima 强调不要从 xpu dispatch 调用 forward_cuda，避免歧义。最终 Gemma4RMSNorm.forward_xpu 自包含。
- store_cache 分发：mingfeima 建议在 store_cache 中添加 xpu dispatch，该改动未包含在本 PR 中，可能移至后续 PR。
- KV cache retrieving 支持：kpham-sgl 提醒需要实现 KV cache retrieving 支持，与 triton_backend.py 对齐。本 PR 通过 k=None, v=None 路径实现了跨层 KV 共享，覆盖了 retrieving 场景。
  - RMSNorm reshape 必要性和改进方向 (design): 当前实现移除显式 reshape，依赖 sgl-kernel-xpu 的 stride 处理。后续可改进 fused_add_rmsnorm。
  - 避免从 xpu 调用 forward_cuda (design): 最终 forward_xpu 自包含，不再调用 forward_cuda。
  - store_cache 中添加 xpu dispatch (design): 该建议未在本 PR 中实现（memory_pool.py 未改动），可能移至后续 PR。
  - 需实现 KV cache retrieving 支持 (design): 本 PR 通过 k=None, v=None 路径实现了跨层 KV 共享，覆盖了 retrieving 场景。

风险与影响

- 风险：
  1. layernorm.py 依赖 sgl-kernel-xpu 的 stride 处理：移除 reshape 后，若底层 kernel 行为变更或处理特定维度出错，可能导致数值错误。GSM8K 验证覆盖有限。
  2. SWA 页表翻译性能风险：每次前向元数据初始化均执行翻译和 int32 类型转换，在大 batch 或长上下文时可能增加延迟。
  3. 融合 routing kernel 精度一致性：新增 gemma_routing_post_topk 需确保与旧路径数值一致，当前仅验证 E2B 模型。
  4. 测试覆盖不足：只有 E2B 模型的烟囱测试，缺少 E4B、31B、26B-A4B 的 CI 测试，也缺少对 SWA 长上下文 decode 的显式准确性验证。
  5. cross-layer KV 共享路径：非 SWA 层的共享逻辑可能与其他后端（如 Triton）行为存在差异，需更多交叉验证。 - 影响：影响范围：主要影响 Intel XPU 用户，使其能够使用 Gemma 4 系列模型进行推理，无需降级至其他后端。影响程度：中等。代码改动主要集中在新后端适配，不改变现有 CUDA/ROCm/Triton 后端的行为。新增约 350 行代码，维持了模块化设计。测试：新增 E2B 测试并注册 CI，为后续模型适配提供了范例。 - 风险标记：核心路径变更，依赖底层 kernel，测试覆盖有限

关联脉络

- PR #27321 docs(cookbook): restore Gemma 4 transformers commit pin: Gemma 4 模型文档与本 PR 的 XPU 支持互补，为用户提供部署指南。
- PR #27316 fix(attn): delegate init_mha_chunk_metadata in HybridLinearAttnBackend: 混合注意力修复与本 PR 的 SWA 支持在混合注意力场景下相关。
- PR #27287 docs(cookbook): add Docker install option for Gemma 4: Gemma 4 部署文档，与本 PR 的 XPU 支持形成完整工具链。