PR #48858 完整报告

vllm-project/vllm

[Model] Add Hopper FA4 relative attention for Inkling

合并时间：2026-07-17 02:26

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48858

执行摘要

- 一句话：为 Inkling 模型在 Hopper 上添加 FA4 score_mod 相对注意力路由
- 推荐动作：值得精读，尤其是 _get_score_mod 中通过 cutlass.cute.jit 在 score_mod 内实现相对偏置的设计模式，以及如何通过单一入口函数根据架构路由到不同 FA4 后端。这种模式可复用于其他模型的多架构适配。

功能与动机

Inkling 模型的相对注意力实现依赖 tml-fa4 sheared-bias 接口，该接口仅 Blackwell 架构支持。在 Hopper 上会产生不兼容的 SM90 调用签名和 split-KV 约束。因此需要在 Hopper 上使用常规 FA4 score_mod 机制实现相对偏置，以支持 Inkling 模型在 Hopper 上的推理。

实现拆解

1. 架构检测函数 _use_sheared_bias：在 vllm/models/inkling/nvidia/ops/fa4_rel_attention.py 中新增带缓存的函数，根据 GPU 计算能力（major 版本）返回是否使用 sheared-bias。Blackwell（SM100/SM110）返回 True，其他（包括 Hopper SM90）返回 False。
2. score_mod 生成函数 _get_score_mod：新增同样带缓存的函数，返回一个 JIT 编译的 score_mod_rel_bias 可调用，通过 cutlass.cute.jit 装饰。该函数利用 aux_tensors[0]（即 rel_logits）计算相对距离并取出对应偏置值，实现式 $(1/head_dim) \cdot Q \cdot K + rel_bias$。
3. inkling_fa4_rel_attention 路由修改：在原有函数中，先 rel_logits = rel_logits.contiguous()，然后根据 _use_sheared_bias() 选择导入不同的 flash_attn_varlen_func 并传递对应参数。Blackwell 路径传入 rel_bias，Hopper 路径传入 score_mod 和 aux_tensors。
4. 测试覆盖：在 tests/models/inkling/test_fa4_rel_attention.py 中新增 test_sheared_bias_architecture_selection 参数化测试（SM90/100/110/120），验证路由逻辑；新增 test_score_mod_relative_attention 在 Hopper 模拟下调用 score_mod 路径并与 PyTorch 参考对比。同时修复了已有测试中 num_splits 参数的传递。

关键文件：

- vllm/models/inkling/nvidia/ops/fa4_rel_attention.py（模块 模型实现；类别 source；类型 core-logic；符号 _use_sheared_bias, _get_score_mod, score_mod_rel_bias, inkling_fa4_rel_attention）：核心路由逻辑，新增 _use_sheared_bias 和

_get_score_mod 函数，修改 inkling_fa4_rel_attention 以支持 Hopper 的 score_mod 路径。

- tests/models/inkling/test_fa4_rel_attention.py（模块 测试；类别 test；类型 test-coverage；符号 test_sheared_bias_architecture_selection, test_score_mod_relative_attention）：新增架构选择测试和 score_mod 数值测试，确保 Hopper 路径的正确性。

关键符号：_use_sheared_bias, _get_score_mod, score_mod_rel_bias, inkling_fa4_rel_attention

关键源码片段

vllm/models/inkling/nvidia/ops/fa4_rel_attention.py

核心路由逻辑，新增 _use_sheared_bias 和 _get_score_mod 函数，修改 inkling_fa4_rel_attention 以支持 Hopper 的 score_mod 路径。

```
# vllm/models/inkling/nvidia/ops/fa4_rel_attention.py
from functools import cache
from collections.abc import Callable

@cache
def _use_sheared_bias() -> bool:
    """返回当前 GPU 是否应使用 sheared-bias 路径（Blackwell SM100/SM110）。"""
    capability = current_platform.get_device_capability()
    return capability is not None and capability.major in (10, 11)

@cache
def _get_score_mod(rel_extent: int) -> Callable:
    """返回一个 jit 编译的 score_mod，从 aux_tensors 读取相对偏置。"""
    import cutlass.cute as cute
    from cutlass.cute import Float32
    from vllm.vllm_flash_attn.cute.seqlen_info import SeqlenInfoQK

    @cute.jit
    def score_mod_rel_bias(
        scores: cute.TensorSSA,
        b_idx: cute.TensorSSA,
        h_idx: cute.TensorSSA,
        q_idx: cute.TensorSSA,
        kv_idx: cute.TensorSSA,
        seqlen_info: SeqlenInfoQK,
        aux_tensors: list[cute.Tensor],
    ) -> cute.TensorSSA:
        rel_logits = aux_tensors[0] # shape: (total_tokens, num_heads, rel_extent)
        seqlen_local_offset = seqlen_info.seqlen_k - seqlen_info.seqlen_q
        rel_dist = (q_idx + seqlen_local_offset) - kv_idx
        global_q_idx = seqlen_info.offset_q + q_idx
        # 将相对距离限制在 [0, rel_extent-1] 内，超出范围则为 0
```

```
        rel_dist_0 = rel_dist[0]
        rel_idx = rel_dist_0 if rel_dist_0 >= 0 else 0
        rel_idx = rel_idx if rel_idx < rel_extent else (rel_extent - 1)
        rel_bias = rel_logits[global_q_idx[0], h_idx[0], rel_idx]
        rel_bias = Float32(rel_bias) if rel_dist_0 == rel_idx else Float32(0.0)
        return scores + rel_bias

    return score_mod_rel_bias

def inkling_fa4_rel_attention(...):
    rel_logits = rel_logits.contiguous()
    if _use_sheared_bias():
        from vllm.third_party.tml_fa4 import flash_attn_varlen_func
        bias_kwargs = {"rel_bias": rel_logits}
    else:
        from vllm.vllm_flash_attn.cute import flash_attn_varlen_func
        bias_kwargs = {
            "score_mod": _get_score_mod(rel_extent),
            "aux_tensors": [rel_logits],
        }
    ret = flash_attn_varlen_func(
        ...,
        num_splits=num_splits,
        out=out,
        **bias_kwargs,
    )
```

评论区精华

该 PR 无实质性的 review 讨论。仅有 claude[bot] 自动评论提示手动 code review，但未触发。PR 由作者直接合并。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于 Hopper 路径的数值正确性未经实际 Hopper 硬件验证（作者提到 GB200 上测试，但 Hopper 未测试，依赖 CI 的 H200 验证）。Blackwell 路径未修改，回归风险极低。另外，aux_tensors 传递 rel_logits 给 score_mod 的语义依赖 Cutlass 和 vLLM Flash Attention 版本，版本升级可能引入兼容性问题。
- 影响：影响范围有限，仅涉及 Inkling 模型的相对注意力计算路径。用户：Inkling 模型在 Hopper 上首次获得相对注意力支持，性能与正确性由新路径保证。系统：无全局影响。团队：需维护两套后端路由，但代码抽象清晰，扩展性好。
- 风险标记：Hopper 硬件未测试，依赖 Cutlass/FA4 版本兼容性，仅 Python 层变更，内核无影响

关联脉络

- PR #48822 [Model] Add PW CUDA graph support for Inkling [2/N]: 同为 Inkling 模型系列变更，先后引入 CUDA 图和 Hopper FA4 支持，形成完整的多架构适配链条。
- PR #46275 [ROCm][Perf][DSV4] Enable split sparse decode on gfx942: 虽然无关 Inkling，但展示了项目内在不同架构上适配注意力计算的一般模式，可作横向参考。