PR #51860 完整报告

vllm-project/vllm

[ROCm][K3] Dequantize the fp8 decode query for MLA backends without quant-query support - TRITON_MLA

合并时间：2026-08-13 00:40

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51860

执行摘要

- 一句话：修复 Kimi-K3 fp8 KV 解码断言，为 TRITON_MLA 增加 query 反量化回退
- 推荐动作：值得精读：改动虽小，但涉及 backend 能力契约与数值 dtype 保持两个容易踩坑的点，以及 ROCm/TRITON_MLA 生态下的实测方法（作者给出了可复现 serve 命令与 gsm8k 结果）。关注点：_q_scale 与 _q_scale_inv 的一致性如何保证；后续是否值得把这种 fallback 下沉到后端或补充自动化覆盖。

功能与动机

PR body 明确说明：在 DSpark 投机解码中启用 fp8 KV cache dtype 时触发 assert 错误，需要最小修复以解锁该路径（minimal fix to unblock this path）。TRITON_MLA 是 ROCm 上唯一支持非因果多 token 块的 MLA backend，但它 dequantizes fp8 KV on load and takes a bf16 query，因此 Kimi-K3 _decode_concat_cache 对 supports_quant_query_input 的断言使 fp8 KV 无法与 DSpark 组合使用，启动即报 RuntimeError: Worker failed with error 'Kimi-K3 fp8 KV cache decode requires a backend that accepts an fp8 (quantized) query input.'。修复后实测 gfx950 x8 TP8 可正常服务，gsm8k 1319q greedy 达 94.4%，acceptance 3.40。

实现拆解

1. 变更入口：vllm/models/kimi_k3/nvidia/mla.py 的 _decode_concat_cache 方法，删除 is_quantized_kv_cache(self.kv_cache_dtype) 分支内对 self.impl.supports_quant_query_input 的硬断言（该断言在 TRITON_MLA 下启动即失败）。
2. 核心逻辑调整：fp8 路径仍以原参数调用 fused_mla_decode_q_concat_kv_cache_insert（传入 q_scale_inv、cache_scale_inv），但返回值先存入 mqa_q；随后判断 not self.impl.supports_quant_query_input，为 True 时执行 (mqa_q.to(torch.float32) * self._q_scale).to(ql_nope.dtype)，撤销 query 量化并恢复原始 dtype，再统一返回 mqa_q。
3. dtype 保持：最终版本特意在 fp32 上完成乘法再一次性 cast 回 ql_nope.dtype，避免 fp32 标量 _q_scale 把结果提升到 fp32，破坏下游期望 bf16 query 的 backend（由 Copilot review 提出并修正）。

4. 影响面控制：supports_quant_query_input=True 的 backend（当前 NVIDIA 各 K3 backend、ROCM_AITER_MLA）走完全相同的 fused 调用，新分支不生效；非 DSpark 的 fp8 + TRITON_MLA 请求不进入 _decode_concat_cache，走通用 MLA 流程，由 mla_attention.py 按能力决定是否量化 query。
5. 测试配套：未新增自动化测试，作者提供可复现 serve 命令及 gsm8k 手动验证结果（94.4% / 94.7%）。

关键文件：

- vllm/models/kimi_k3/nvidia/mla.py（模块 模型层；类别 source；类型 core-logic；符号 _decode_concat_cache）：Kimi-K3 MLA 解码路径的核心文件，_decode_concat_cache 中的断言导致 ROCm DSpark + fp8 KV 组合启动失败；本 PR 在此移除断言并增加 query 反量化回退。

关键符号：_decode_concat_cache

关键源码片段

vllm/models/kimi_k3/nvidia/mla.py

Kimi-K3 MLA 解码路径的核心文件，_decode_concat_cache 中的断言导致 ROCm DSpark + fp8 KV 组合启动失败；本 PR 在此移除断言并增加 query 反量化回退。

```
def _decode_concat_cache(
    self,
    ql_nope: torch.Tensor,
    q_pe: torch.Tensor,
    kv_c_normed: torch.Tensor,
    k_pe: torch.Tensor,
    positions: torch.Tensor | None,
    cos_sin_cache: torch.Tensor | None,
    slot_mapping: torch.Tensor,
) -> torch.Tensor:
    """Fused decode query-concat + latent cache insert, dispatched by cache
    dtype (same policy as prefill: fp8 cache -> fp8 query)."""
    if self.kv_cache_dtype == "fp8_ds_mla":
        # DeepSeek 风格 fp8 MLA（uint8 cache）走独立分支，不涉及本次改动。
        cache = self.kv_cache
        if cache.dtype != torch.uint8:
            cache = cache.view(torch.uint8)
        return fused_mla_decode_q_concat_kv_cache_insert(
            ql_nope, q_pe, kv_c_normed, k_pe, cache, slot_mapping,
            ds_mla=True, positions=positions, cos_sin_cache=cos_sin_cache)

    if is_quantized_kv_cache(self.kv_cache_dtype):
        cache = self.kv_cache
        if cache.dtype != torch.float8_e4m3fn:
            cache = cache.view(torch.float8_e4m3fn)
        # 照常执行 fused fp8 query 拼接 + cache 插入内核，返回量化后的 query。
        mqa_q = fused_mla_decode_q_concat_kv_cache_insert(
```

```
            ql_nope, q_pe, kv_c_normed, k_pe, cache, slot_mapping,
            q_scale_inv=self._q_scale_inv,
            cache_scale_inv=self._k_scale_inv,
            positions=positions, cos_sin_cache=cos_sin_cache)
        if not self.impl.supports_quant_query_input:
            # TRITON_MLA 等 backend 加载时反量化 fp8 KV，只接收 bf16 query，
            # 这里撤销上面的 query 量化：先转 fp32 乘 _q_scale 还原真实值，
            # 再一次性 cast 回 ql_nope.dtype，避免 fp32 标量提升结果 dtype。
            mqa_q = (mqa_q.to(torch.float32) * self._q_scale).to(ql_nope.dtype)
        return mqa_q

    # bf16 cache 的普通路径：query 不做量化，直接返回 fused 内核结果。
    return fused_mla_decode_q_concat_kv_cache_insert(
        ql_nope, q_pe, kv_c_normed, k_pe, self.kv_cache, slot_mapping,
        positions=positions, cos_sin_cache=cos_sin_cache)
```

评论区精华

1. Copilot 指出最初实现 mqa_q.to(ql_nope.dtype) * self._q_scale 会因 _q_scale 是 fp32 tensor 而把结果提升回 fp32，可能破坏期望 bf16/fp16 query 的 backend 并带来额外开销，建议 cast _q_scale 或 cast 乘积。
2. njhill 认为 supports_quant_query_input 是 MLA backend 的能力契约，直接属性访问即可，不必用 getattr(..., False)。
3. njhill 在子线程追问作者是否处理了 Copilot 的意见，dllehr-amd 回复 "Jumped the gun on this. A simple cast should work here"。
4. 最终提交（Address review）落实了 fp32 计算 + 单次 cast 回 ql_nope.dtype 的修复，并改为直接属性访问。评审结论：dllehr-amd 批准，njhill 致谢，无未解决疑虑。

- 反量化后 query 的 dtype 被 fp32 标量提升 (correctness): 改为 (mqa_q.to(torch.float32) * self._q_scale).to(ql_nope.dtype)，在 fp32 计算后一次性 cast 回 ql_nope.dtype。
- 使用 getattr 还是直接属性访问 (style): 接受 njhill 建议，改为直接属性访问 self.impl.supports_quant_query_input。
- Copilot 评论是否已处理 (question): 后续提交（Address review）落实了 fp32 计算 + 单次 cast 的修复。

风险与影响

- 风险：
  1. 数值精度风险：反量化依赖 _q_scale 与 fused kernel 内 _q_scale_inv 互为倒数，任何 scale 不匹配都会引入精度损失；作者报告 gsm8k 94.4%，但未与 bf16 KV 基线对比。
  2. dtype 风险：_q_scale 为 fp32，最终版本虽在最后 cast 回 ql_nope.dtype，但涉及尾数截断，对后续数值敏感路径有潜在影响。
  3. 覆盖缺口：没有自动化测试，supports_quant_query_input=False 分支的回归（如未来 backend 能力变化）不会被 CI 捕获。

4. 波及面：修改位于 vllm/models/kimi_k3/nvidia/mla.py 的核心 decode 热路径，虽然对支持量化 query 的 backend 是 no-op，但属于共享控制流调整。 - 影响：用户影响：解锁 ROCm（gfx950）上 Kimi-K3 + DSpark 投机解码 + fp8 KV cache 的组合，此前直接启动失败；非 DSpark 的 fp8 + TRITON_MLA 走通用 MLA 流程不受影响。系统影响：NVIDIA 各 K3 backend 行为不变（该分支不可达），ROCM_AITER_MLA 同样不受影响。团队影响：改动小且集中，API 契约（supports_quant_query_input）未变，但为后续 backend 能力差异的处理提供了一个"先执行后反量化"的 fallback 样板。 - 风险标记：核心解码路径变更，缺少自动化测试，dtype 精度敏感，fallback 依赖 backend 能力契约

关联脉络

- PR #50654 [ROCm][Perf] Kimi-K3 Fused kernel for KDA decode: 同为 ROCm 上 Kimi-K3 的注意力 / 解码内核优化，是 K3 ROCm 支持线路的一部分，与本次 MLA decode 路径改动属于同一硬件平台的功能演进。
- PR #51831 [Model] Support R3 capture with DeepGEMM MegaMoE: 涉及 Kimi K3 模型执行路径的演进与新特性支持，说明 K3 模型近期有系统性改动。
- PR #51843 [Bugfix] Disable fine-grained prefix-cache hits for incompatible hybrid KV layouts: 同为 kimi/K3 相关 KV 布局兼容性修复，说明 K3 的 KV 路径近期有系统性调整，本 PR 是其中一环。