PR #34261 完整报告

sgl-project/sglang

[AMD] Restore K3 MLA verify kernel path blocked by can_handle() guard

合并时间：2026-08-10 18:31

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34261

执行摘要

- 一句话：恢复 K3 MLA 验证内核路径，解耦守卫
- 推荐动作：该 PR 值得精读，它提供了一个典型的共享守卫耦合导致功能失效的案例，并展示了通过复制 + 微调来解耦的务实方案。关注点：后续可考虑提取公共门控模板或参数化配置，减少重复；同时建议补充针对 verify_mla.can_handle 的单元测试，防止未来回归。

功能与动机

PR body 明确指出：PR #29677 给 can_handle() 添加了拒绝 q_head_dim != v_head_dim 的守卫，而 MLA 恰好属于该情形，导致 verify_mla_fwd 不再被启用，新内核从未运行，推理性能回退。本 PR 的目标是恢复 K3 的 MLA verify 路径。

实现拆解

1. 移除共享导入：在 verify_mla.py 中停止从 verify_splitkv 导入 can_handle，仅保留 _AMD_LAUNCH_KWARGS 导入。
2. 新增专用门控：在 verify_mla.py 模块内定义独立的 can_handle 函数，逐行镜像 verify_splitkv 的逻辑，但有意删除 head-dim 相等性检查（MLA 场景下 q_head_dim != v_head_dim 是合法情况）。
3. 保持保守门控：新函数继续拒绝 sliding window、sinks、logit cap、非因果等特性，并只接受常量扩展长度（通过纯 shape 检查），确保在 HIP graph 捕获中无 host sync，避免 stream capture 失败。
4. 测试配套：本 PR 未新增单元测试，依赖 AMD CI（run-ci 标签）覆盖 K3 路径验证，由于变更仅影响特定硬件路径，风险可控。

关键文件：

- python/sglang/kernels/ops/attention/verify_mla.py（模块 注意力内核；类别 source；类型 core-logic；符号 can_handle）：唯一变更文件，新增 verify_mla 专用 can_handle 函数并移除对 verify_splitkv.can_handle 的依赖，直接恢复 Kimi-K3 MLA 验证内核的启用条件。

关键符号：can_handle

关键源码片段

python/sglang/kernels/ops/attention/verify_mla.py

唯一变更文件，新增 verify_mla 专用 can_handle 函数并移除对 verify_splitkv.can_handle 的依赖，直接恢复 Kimi-K3 MLA 验证内核的启用条件。

```
# verify_mla.py 中新增的 verify_mla 专用门控函数
def can_handle(
    q_extend, k_extend, v_extend, k_buffer, v_buffer,
    qo_indptr, kv_indptr, kv_indices, custom_mask, is_causal,
    mask_indptr, max_len_extend, sliding_window_size=-1,
    sinks=None, logit_cap=0.0, xai_temperature_len=-1,
):
    """判断 MLA split-KV verify 路径能否处理该问题。
    与 verify_splitkv.can_handle 逐行对齐，但有意移除 q_head_dim != v_head_dim
    的拒绝分支 —— MLA 的 q_head_dim 恒不等于 v_head_dim。
    保守原则：任何未显式支持的特性都返回 False，由调用方回退到 baseline。

    注意：custom_mask 的值不会被检查（在 HIP graph 捕获中读取会触发
    host sync），因此必须依赖调用方保证 topk == 1 时才启用此路径。
    """
    # 不支持滑动窗口、sinks、logit cap、温度缩放等特性
    if sinks is not None:
        return False
    if sliding_window_size is not None and sliding_window_size > 0:
        return False
    if logit_cap and logit_cap > 0:
        return False
    if xai_temperature_len is not None and xai_temperature_len > 0:
        return False
    if not is_causal:
        return False

    # q 布局必须为 [tokens, H_Q, D]（head 维度由 power-of-2 padding 处理）
    if q_extend.dim() != 3 or k_extend.dim() != 3 or v_extend.dim() != 3:
        return False

    # GQA 分组必须整除
    h_q = q_extend.shape[1]
    h_kv = k_extend.shape[1]
    if h_kv == 0 or h_q % h_kv != 0:
        return False

    # head 维度必须与 buffer 匹配
    if k_buffer.shape[1] != h_kv or v_buffer.shape[1] != h_kv:
        return False
    if q_extend.shape[2] != k_extend.shape[2]:
        return False
    if q_extend.shape[2] != k_buffer.shape[2]:
        return False
```

```
    if v_extend.shape[2] != v_buffer.shape[2]:
        return False

    # 关键：此函数绝不能读取张量 * 值 *（无 .item() / .cpu()），
    # 因为 target-verify 在捕获的 CUDA/HIP graph 内运行，
    # device->host 同步会触发 hipErrorStreamCaptureUnsupported。
    # 因此只基于静态 shape / dtype / python 标量做门控。
    bs = qo_indptr.shape[0] - 1
    if bs < 1:
        return False

    # max_len_extend 必须是已知的正 python int
    # （verify 路径中为 server_args.speculative_num_draft_tokens）
    # topk=1 时每序列 extend 长度恒等于 num_draft_tokens，
    # 等于 max_len_extend，因此行 tile mask 精确匹配，无需检查值。
    try:
        mle = int(max_len_extend)
    except (TypeError, ValueError):
        return False
    if mle < 1:
        return False

    # 打包的 extend 张量必须恰好包含 bs * max_len_extend 行
    # （常量 extend 长度），纯 shape 检查无同步，
    # 拒绝 ragged / 变长 extend 批次，使之回退 baseline。
    if q_extend.shape[0] != bs * mle:
        return False
    return True
```

评论区精华

PR 无评论和 review 评论，唯一审核来自 HaiShaw（APPROVED，body 为 'AMD gated.'）。核心设计讨论集中在 PR body 中：究竟是让 verify_mla 复用 splitkv 的 can_handle 并通过参数豁免 head-dim 检查，还是直接复制一份独立实现。最终选择了复制，以避免改动 splitkv 逻辑影响其他 MHA 路径。

- verify_mla 复用 verify_splitkv.can_handle 导致 MLA 被误拒 (correctness): 采用复制 can_handle 并移除 head-dim 检查的方案，避免改动 splitkv 逻辑；HaiShaw 审核批准（AMD gated）。

风险与影响

- 风险： 1) 逻辑重复风险：verify_mla.can_handle 与 verify_splitkv.can_handle 为两份独立副本，未来若 splitkv 修复 bug 或新增特性，verify_mla 不会自动同步，可能产生行为漂移。 2) 缺少测试覆盖：PR 未增加任何单元测试，仅依赖 AMD CI 的隐式覆盖，若未来引入类似回归，难以提前捕获。 3) 影响面局限：变更仅启用 verify_mla 路径，若内核本身存在未暴露的缺陷（如数值精度），可能在新路径上产生静默错误，但 PR body 已确认只影响 K3/mi355。

- 影响：对用户：恢复 Kimi-K3 在 AMD 平台使用 DSpark 时的 MLA verify 加速，消除因守卫导致的性能回退。对系统：门控逻辑在 verify_mla.py 内自包含，不影响其他注意力路径。对团队：需要维护两份相似的门控函数，增加轻微维护成本，但避免了对 splitkv 行为的副作用。
- 风险标记：仅 AMD 路径，缺少测试覆盖，门控逻辑重复维护

关联脉络

- PR #33981 Add triton MLA verify kernel for Kimi-K3 DSpark: 引入 verify_mla_fwd 内核，本 PR 修复其被后续守卫禁用的回归。
- PR #29677 Add head-dim guard to verify_splitkv can_handle(): 新增 q_head_dim != v_head_dim 拒绝分支，无意中禁用了 MLA 路径。