PR #35197 完整报告

sgl-project/sglang

fix(kernel) Fix Helion small-token prefill bug

合并时间：2026-08-21 07:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35197

执行摘要

- 一句话：修复 Helion KDA 短 prefill 形状处理与解码维度校验
- 推荐动作：值得精读。本 PR 揭示了 JIT kernel trace 特化（size-one 维度常量折叠）导致缓存污染的真实案例，以及通过「路由到 Triton fallback」规避的通用设计模式；同时 wrapper 入口做契约对齐（裁剪 padding）的做法也可借鉴。重点关注 kda_prefill.py 的 T=1 fallback 与 kda_decode.py 的输入校验策略。

功能与动机

PR body 明确指出 Helion KDA 后端的三个问题：短 prefill 时 Q/K/V 只含 active tokens 而 gate/beta 保持 CUDA-graph bucket padding，从 padded gel 推导 token 数导致非法 reshapes；若首个 prefill 请求 T=1，PyTorch 会在生成的 Helion trace 中特化 size-one 维度，影响后续调用；decode 在 key/value 维度非 2 的幂时使用 padded lanes 而无 mask。这些会导致短 prompt 场景下结果错误或崩溃，需要对齐 Triton chunk_kda 契约并明确拒绝非法配置。

实现拆解

按 4 个步骤拆解：

1. prefill入口形状校验与裁剪（python/sglang/kernels/ops/attention/helion/kda_prefill.py）：在 chunk_kda 的公开入口处，先要求 g、beta 的 token 维度不小于 q.shape[1]，然后统一裁剪为 [:, :num_tokens]，保证与 Triton chunk_kda 的契约一致，消除从 padded tensor 推导 token 数导致的非法 reshape。
2. T=1 请求路由到 Triton（同文件）：当 num_tokens == 1 时直接返回 triton_chunk_kda(...)，避免 PyTorch tracing 常量折叠 size-one 维度后与更长输入共享内核缓存条目，防止首个单 token prefill 污染后续 Helion 调用；T>1 路径保持不变。
3. decode 非 2 幂维度硬校验（python/sglang/kernels/ops/attention/helion/kda_decode.py）：新增 _is_power_of_two 帮助函数，并在 validate_packed_decode_inputs 中对 initial_state 的 key、value head 维度做 2 的幂检查，不满足直接抛 ValueError；Kimi linear/K3 的 head dim 均为 128，不受影响。
4. 回归测试补强（test/registered/kernels/ops/attention/test_kda_helion.py）：新增 test_single_token_prefill_does_not_poison_later_shapes 与 test_prefill_ignores_padded_gate_rows 两个参数化测试（fixed/varlen 两种布局），分别

验证 T=1 不污染后续 shape、gate/beta 的 padding 大值行被忽略；PR body 另给出覆盖 19 种长度、3 种 head 数、2 种布局、2 种精度的 216 组 Helion/Triton 对比 sweep，全部通过。

无配置或部署配套改动。

关键文件：

- python/sglang/kernels/ops/attention/helion/kda_prefill.py（模块 KDA 内核；类别 source；类型 core-logic；符号 chunk_kda）：核心修复文件：在 chunk_kda 公开入口裁剪 gate/beta 的 padded 行，并将 T=1 请求路由到 Triton，解决短 prefill 的 token 数推导与 size-one trace 特化问题。
- python/sglang/kernels/ops/attention/helion/kda_decode.py（模块 KDA 内核；类别 source；类型 core-logic；符号 _is_power_of_two, validate_packed_decode_inputs）：为 decode 输入校验新增 _is_power_of_two，拒绝非 2 的幂 key/value head 维度，避免 padded lanes 无 mask 时的静默错误。
- test/registered/kernels/ops/attention/test_kda_helion.py（模块 内核测试；类别 test；类型 test-coverage；符号 test_single_token_prefill_does_not_poison_later_shapes, test_prefill_ignores_padded_gate_rows, run）：新增两个参数化回归测试，覆盖 T=1 trace 污染与 padding gate 行忽略两个关键修复点，是本次改动正确性的主要保障。

关键符号：chunk_kda, validate_packed_decode_inputs, _is_power_of_two, test_single_token_prefill_does_not_poison_later_shapes, test_prefill_ignores_padded_gate_rows

关键源码片段

python/sglang/kernels/ops/attention/helion/kda_prefill.py

核心修复文件：在 chunk_kda 公开入口裁剪 gate/beta 的 padded 行，并将 T=1 请求路由到 Triton，解决短 prefill 的 token 数推导与 size-one trace 特化问题。

```
# helion/kda_prefill.py —— Helion KDA prefill 公开入口修复
# 修复点 1：短 prefill 时 gate/beta 可能被 padding 到 CUDA-graph bucket 大小，
# 从 padded tensor 推导 token 数会产生非法 reshape，因此在入口按 q 的真实
# token 数裁剪 gate/beta，与 Triton chunk_kda 的契约保持一致。

def chunk_kda(
    q, k, v, g, beta, scale=1.0,
    initial_state=None, initial_state_indices=None,
    use_qk_l2norm_in_kernel=False,
    cu_seqlens=None, A_log=None, dt_bias=None,
    lower_bound=None, output_intermediate_states=False,
):
    # 公开入口必须带索引化的初始状态池
    if initial_state is None or initial_state_indices is None:
        raise ValueError("KDA prefill requires an indexed initial-state pool")

    num_tokens = q.shape[1]
```

```
    # gate/beta 必须覆盖全部 q token，再裁掉 padding 行
    if g.shape[1] < num_tokens or beta.shape[1] < num_tokens:
        raise ValueError("g and beta must cover every q token")
    g = g[:, :num_tokens]
    beta = beta[:, :num_tokens]

    # 修复点 2：T=1 的首个 prefill 会让 PyTorch tracing 常量折叠 size-one 维度，
    # 生成的 Helion 内核可能与更长输入共享缓存条目，导致后续调用特化错误。
    # 因此把 T=1 固定路由到 Triton 实现，T>1 仍走 Helion，性能不受影响。
    if num_tokens == 1:
        return triton_chunk_kda(
            q=q, k=k, v=v, g=g, beta=beta, scale=scale,
            initial_state=initial_state,
            initial_state_indices=initial_state_indices,
            use_qk_l2norm_in_kernel=use_qk_l2norm_in_kernel,
            cu_seqlens=cu_seqlens, A_log=A_log, dt_bias=dt_bias,
            lower_bound=lower_bound,
            output_intermediate_states=output_intermediate_states,
        )

    q = q.contiguous()
    k = k.contiguous()
    # 后续走 Helion chunk 计算路径，保证 T>1 的既有行为不变
```

python/sglang/kernels/ops/attention/helion/kda_decode.py

为 decode 输入校验新增 _is_power_of_two，拒绝非 2 的幂 key/value head 维度，避免 padded lanes 无 mask 时的静默错误。

```
# helion/kda_decode.py —— decode 输入校验修复
# 修复点 3：decode 使用 padded 的 hl.arange lanes 且不带 mask，
# 当 key/value head 维度不是 2 的幂时 lane 映射可能越界或错位，
# 因此在入口直接拒绝这类配置，避免静默错误。

def _is_power_of_two(value: int) -> bool:
    # 标准 2 的幂判定：大于 0 且二进制表示中只有一位为 1
    return value > 0 and value & (value - 1) == 0


def validate_packed_decode_inputs(mixed_qkv, a, ..., initial_state):
    # 已有的连续性等校验 ...
    HV, V, K = initial_state.shape[-3:]
    if not _is_power_of_two(K) or not _is_power_of_two(V):
        raise ValueError(
            "Helion KDA decode requires power-of-two key and value head "
            f"dimensions (got K={K}, V={V})."
        )
    if a.shape[1] != HV * K:
        raise ValueError(
            f"`a` must have shape [B, HV*K] with HV={HV}, K={K} "
```

```
        )
    # ... 其余输入校验
```

评论区精华

本 PR 没有可见的 review comment，但两位维护者 zcnrex 与 yhyang201 均 APPROVED。从 commit 演进看，*decode* 部分经历了「Fix Helion shape handling → Simplify *decode* shape handling → Keep *decode* config unchanged → Minimize *decode* diff」四轮收敛，最终只保留 2 的幂校验逻辑，说明评审中要求 *decode* 改动最小化。另外作者多次触发 /rerun-failed-ci，PR Test (Extra) 曾失败，主测试最终通过。

- *decode* 改动范围收敛 (design): 最终只保留 2 的幂校验（+9 行），*decode* 原有逻辑零改动，两位 maintainer 均 APPROVED。
- CI 稳定性 (other): 无遗留问题。

风险与影响

- 风险： 1) Helion prefill 公开入口现在对 *gate*/*beta* 覆盖不足直接抛异常，调用方若依赖旧行为需要同步调整； 2) T=1 走 Triton、T>1 走 Helion，路径切换可能引入数值或行为不一致，依赖新增测试与 sweep 保障； 3) *decode* 硬性拒绝非 2 的幂 key/value head 维度，若未来出现 head dim 非 2 的幂的自定义模型，需先补充 mask 支持； 4) 改动集中于 Helion 专属后端，不影响默认 CUDA 路径。
  - 影响：影响范围限于启用 Helion KDA 后端的场景（如 Kimi 系列模型在 Helion 硬件上）。短 prefill 与首个 token 的正确性显著提升，*decode* 对非法维度从静默错误变为清晰报错；与 Triton 后端的契约对齐降低了 wrapper 维护成本。测试矩阵覆盖多长度、多 head 数、多精度，团队回归信心增强。对默认后端用户无影响。
  - 风险标记：Helion 后端专属修复，T=1 走 Triton fallback, *decode* 硬性拒绝非 2 幂维度，CI 多次重跑

关联脉络

- PR #35689 Skip empty linear-attention state buffers in PD transfer: 同属 linear-attention/KDA 状态缓冲区边界 bugfix，都是对状态 shape 与传输边界的静默错误补强。
- PR #35412 [Fix] Land the *decode* mamba checkpoint depth on the tree page under DCP: 同为 KDA/linear-attention 解码状态边界修复，涉及状态 shape 与 batch 结果处理的正确性。