PR #42072 完整报告

vllm-project/vllm

[ROCm] Restore fast top_k_per_row kernels for sparse MLA when topk_tokens=2048

合并时间：2026-05-16 03:02

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42072

执行摘要

- 一句话：恢复稀疏 MLA 中 topk_tokens=2048 的快速 C++ 内核路径
- 推荐动作：值得精读。这是一个典型的“性能回归修复 + 架构清理”组合 PR，展示了如何在不影响通用性的前提下为常见配置恢复专用加速路径。_topk_indices_prefill/_topk_indices_decode 的分发模式可复用。

功能与动机

DeepSeek-V3.2 使用 topk_tokens==2048，正好是专用 C++ 内核设计的目标值。PR #40871 移除了该快速路径，导致 TPOT 显著退化（内部基准显示 ~26 ms vs ~18 ms）。需要将 2048 这一常见配置路由回专用内核以恢复性能。

实现拆解

1. 定义快速路径集合：在 rocm_aiter_mla_sparse.py 中新增 _TOPK_FAST_PATH_VALUES = frozenset({2048})，明确哪些 topk_tokens 值使用专用 C++ 内核。
2. 新增调度函数 _topk_indices_prefill 和 _topk_indices_decode：分别处理 prefill 和 decode 阶段的 top-k 索引计算。若 topk_tokens 在快速路径集合中，则调用 torch.ops._C.top_k_per_row_prefill/torch.ops._C.top_k_per_row_decode；否则回退到 _topk_indices_torch。
3. 修改 prefill 路径调用：将 rocm_aiter_sparse_attn_indexer_native 中 prefill 部分的 topk_indices.copy_(_topk_indices_torch(...)) 替换为 _topk_indices_prefill 调用。
4. 修改 decode 路径调用：同样替换 decode 部分；同时修正 topk_indices 视图切片范围为 [:num_padded_tokens] 以匹配内核写入行数，并将后续 reshape 的参数从 -1 改为 next_n。

关键文件：

- vllm/v1/attention/ops/rocm_aiter_mla_sparse.py（模块 注意力；类别 source；类型 core-logic；符号 _topk_indices_prefill, _topk_indices_decode, _TOPK_FAST_PATH_VALUES）：唯一修改的文件，定义了快速路径集合、新增两个调度函数，并修改了 prefill/decode 路径的调用逻辑。

关键符号：_topk_indices_prefill, _topk_indices_decode

关键源码片段

vllm/v1/attention/ops/rocm_aiter_mla_sparse.py

唯一修改的文件，定义了快速路径集合、新增两个调度函数，并修改了 prefill/decode 路径的调用逻辑。

```
# vllm/v1/attention/ops/rocm_aiter_mla_sparse.py # topk_tokens 值集合，若有专用 fused C++ 内核支持则在此列出 _TOPK_FAST_PATH_VALUES = frozenset({2048}) def_topk_indices_prefill(  logits: torch.Tensor,  topk_tokens: int,  topk_out: torch.Tensor,  cu_seqlen_ks: torch.Tensor,  cu_seqlen_ke: torch.Tensor, ) -> None: """prefill 阶段的 top-k 索引计算。 将 logits.shape[0] 行写入 topk_out；调用方需确保视图大小正确。 """  if topk_tokens in _TOPK_FAST_PATH_VALUES:  # 使用专用 C++ 内核（仅当 topk_tokens == 2048）  torch.ops._C.top_k_per_row_prefill(  logits, cu_seqlen_ks, cu_seqlen_ke, topk_out,  logits.shape[0], logits.stride(0), logits.stride(1), topk_tokens,  )  else:  # 通用回退：使用 torch.topk  topk_out.copy_(_topk_indices_torch(logits, topk_tokens)) def_topk_indices_decode(  logits: torch.Tensor,  topk_tokens: int,  topk_out: torch.Tensor,  seq_lens: torch.Tensor,  next_n: int, ) -> None: """decode 阶段的 top-k 索引计算。 写入 logits.shape[0] == batch_size * next_n 行到 topk_out； 调用方需确保视图大小为 num_padded_tokens。 """  if topk_tokens in _TOPK_FAST_PATH_VALUES:  torch.ops._C.top_k_per_row_decode(  logits, next_n, seq_lens, topk_out,  logits.shape[0], logits.stride(0), logits.stride(1), topk_tokens,  )  else:  topk_out.copy_(_topk_indices_torch(logits, topk_tokens))
```

（注：此处省略原始文件中约 600 行不相关代码，仅展示核心新增部分。）

评论区精华

Gemini-code-assist 在 review 中指出 decode 路径中 topk_indices 视图切片使用 num_decode_tokens 会导致与内核 num_rows = logits.shape[0]（即 num_padded_tokens）不匹配，当 requires_padding=True 时可能造成越界写入及后续 reshape 失败。作者根据反馈修正了切片范围和 reshape 参数。

- decode 路径视图切片越界 (correctness): 作者采纳反馈，将切片范围改为 num_padded_tokens，并修正 reshape 参数。

风险与影响

- 风险：
  - 回归风险：仅在 topk_tokens==2048 时启用快速路径，其他值仍走通用回退，功能正确性由测试保证。
  - 兼容性：仅影响 ROCm 平台的稀疏 MLA 模块，与 NVIDIA 及其他后端无关。
  - 性能：恢复后 TPOT 应回归到 ~18ms 水平，未引入额外开销。
  - 边界条件：decode 路径中 num_padded_tokens 与视图切片的匹配已由 review 修复，但仍需确认所有调用场景（无 padding 等）均正确。
- 影响：

- 用户：DeepSeek-V3.2 用户直接在 TPOT 上获得 ~30% 性能提升。
- 系统：无；改动仅涉及 ROCm 上稀疏 MLA 的一个文件。
- 团队：为 future 支持其他 topk_tokens 值（如 non-2048）的专用内核提供了清晰的分发模式。
- 风险标记：核心路径变更，边界条件修复

关联脉络

- PR #40871 [ROCm][MLA] Support non-2048 top-k tokens via torch fallback: 本 PR 恢复被 #40871 移除的快速路径，二者互为逆向变更。
- PR #42604 DeepSeekV4-Pro enable cuda graph full and piecewise mode: 同一仓库近期与 DeepSeek 稀疏 MLA 相关的性能优化 PR。