PR #41189 完整报告

vllm-project/vllm

[Bugfix] Fix persistent_topk cooperative deadlock at TopK=1024

合并时间：2026-04-30 12:03

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41189

执行摘要

修复了 persistent_topk CUDA kernel 在 TopK=1024、MTP=1 时的协作栅栏死锁问题。通过按实际 vec_size 查询 occupancy、预留每 SM 一个 CTA 作为 headroom，并在超限时回退到 FilteredTopK，确保极端配置下系统不会挂起。

功能与动机

当 TopK=1024 且 MTP=1 时，GPU SM 数不足以同时调度所有 persistent CTA，而每个 kernel 都是 persistent 的，等待协作栅栏时形成死锁。PR body 明确说明："for topk=1024 and MTP=1, this kernel will deadlock since there's no enough SMs to launch the grid and since each kernel is persistent, no kernel will finish."

实现拆解

1. 按实际 vec_size 查询 occupancy：原有代码对所有变体统一使用 vec_size=4 的模板实例查询 occupancy，但实际可能使用 vec_size=2 或 1，导致 occupancy 高估。修改后根据 vec_size 分别查询对应模板实例，并加入错误检查。
2. 引入 headroom 机制：仅在 max_seq_len > RADIX_THRESHOLD（即需要协作栅栏）时生效。occupancy>1 时预留 num_sms 个 CTA，occupancy==1 时预留 1 个 CTA，避免最易死锁场景。
3. 超限回退到 FilteredTopK：若 total_ctas 超过硬件 resident 上限，检查 smem 容量后调用 FilteredTopKRaggedTransform 作为回退，而非直接报错。
4. 代码简化：合并没有必要重复计算的变量，遵循 reviewer 建议。

csrc/topk.cu

核心修复文件，修改了 persistent_topk launch 逻辑，包括 occupancy 查询、headroom 预留和 fallback 路径。

```
// csrc/topk.cu 中的 launch_persistent_topk 函数（关键片段）
// 修复点：按实际 vec_size 查询 occupancy
int occupancy = 1;
cudaError_t occ_err = cudaSuccess;
if (vec_size == 4) {
  occ_err = cudaOccupancyMaxActiveBlocksPerMultiprocessor(
      &occupancy, P::persistent_topk_kernel<TopK, 4>, P::kThreadsPerBlock,
      smem_size);
} else if (vec_size == 2) {
```

```
  occ_err = cudaOccupancyMaxActiveBlocksPerMultiprocessor(
      &occupancy, P::persistent_topk_kernel<TopK, 2>, P::kThreadsPerBlock,
      smem_size);
} else {
  occ_err = cudaOccupancyMaxActiveBlocksPerMultiprocessor(
      &occupancy, P::persistent_topk_kernel<TopK, 1>, P::kThreadsPerBlock,
      smem_size);
}
TORCH_CHECK(occ_err == cudaSuccess,
    "persistent_topk occupancy query failed: ",
    cudaGetErrorString(occ_err));

// 仅在需要协作时才启用 headroom 逻辑
const bool needs_cooperative =
    static_cast<uint32_t>(max_seq_len) > P::RADIX_THRESHOLD;

const uint32_t hw_resident_cap =
    static_cast<uint32_t>(num_sms) * static_cast<uint32_t>(occupancy);
uint32_t max_resident_ctas = hw_resident_cap;
if (needs_cooperative) {
  // 预留 headroom：occupancy>1 时每 SM 留一个 CTA，否则至少留 1 个
  uint32_t headroom = (occupancy > 1) ? static_cast<uint32_t>(num_sms) : 1u;
  if (max_resident_ctas >= headroom + ctas_per_group) {
    max_resident_ctas -= headroom;
  }
}

uint32_t num_groups = std::min(max_resident_ctas / ctas_per_group,
    static_cast<uint32_t>(num_rows));
if (num_groups == 0) num_groups = 1;
uint32_t total_ctas = num_groups * ctas_per_group;

// 若超限则回退到 FilteredTopK
if (needs_cooperative && total_ctas > hw_resident_cap) {
  TORCH_CHECK(max_smem_per_block >= 128 * 1024,
      "persistent_topk would oversubscribe and the FilteredTopK "
      "fallback requires >=128KB smem per block (have ",
      max_smem_per_block, "). total_ctas=", total_ctas,
      " > num_sms*occupancy=", hw_resident_cap, " (TopK=", TopK,
      ", vec_size=", vec_size, ", ctas_per_group=", ctas_per_group,
      ", smem=", smem_size, ").");
  // 调用 FilteredTopKRaggedTransform 作为 fallback
  cudaError_t status =
      vllm::FilteredTopKRaggedTransform<TopK>(...);
  // ...
}
```

评论区精华

> Reviewer LopezCastroRoberto 指出："Headroom logic runs unconditionally. The headroom/oversubscription logic should only be needed when params.max_seq_len > RADIX_THRESHOLD (32K)."
>
> 针对超限场景的建议："I'd fall back to one of the single-CTA kernels available, i.e. FilteredTopKRaggedTransform or top_k_per_row_decode instead of failing with TORCH_CHECK."
>
> gemini-code-assist[bot] 强调："The most critical case for cooperative deadlocks ... If even a single SM is partially occupied ... this kernel will deadlock. At least one CTA slot should be reserved even when occupancy is 1."

风险与影响

- 回退内核性能风险：超限时使用 FilteredTopK 可能比 persistent 版本慢，但 reviewer 已在 B300 上 benchmark 确认可接受。
- 修改范围极小：仅修改 csrc/topk.cu 一个文件，59 行新增，4 行删除，易于审查和回滚。
- 缺少测试覆盖：没有新增测试用例验证死锁修复和回退路径，建议后续补充。

关联脉络

本 PR 与 #41300（DeepSeek bf16 GEMM 优化）同属 DeepSeek 模型推理优化链路，persistent_topk 是 DeepSeek MoE 路由层的核心 kernel。与 #40956（chunk_kda bugfix）同为 CUDA kernel 层的正确性修复。