PR #36684 完整报告

sgl-project/sglang

[AMD] Enable deepseek-v4 topk_transform v2 kernel

合并时间：2026-08-28 13:36

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36684

执行摘要

- 一句话：在 AMD 上启用 DeepSeek-V4 topk v2 内核，长上下文核内提速约 2.7-3 倍
- 推荐动作：值得精读，尤其是 topk_impl.cuh 中 wave64 ballot 的修复和 topk_v2.cuh 中 cluster 路径的条件编译方式，属于典型的 CUDA 内核跨平台移植范本。若团队在 AMD 上运行 DeepSeek-V4 长上下文服务，建议先小流量灰度并留意 TTT 是否影响吞吐目标；E2E 收益主要体现在 TTFT/ITL 而非总耗时，选型时需结合业务指标。

功能与动机

PR body 明确说明：v2 内核用 12 位 key 和寄存器驻留行取代 v1 的 8 位粗粒度 key 与 128KB LDS 候选缓冲，更少候选进入 refinement，且 GPU 不再每 CU 运行一个 block。此前 topk v2 仅在 CUDA 可用（SM120 因 tcgen05/TMEM 缺失而禁用），AMD HIP 路径强制关闭。本 PR 的目标是把这一性能收益带到 AMD 平台，并补上平台 CI 验证。

实现拆解

1. 配置入口切换：在 python/sglang/srt/server_args.py 的 _handle_model_specific_adjustments 中，将 HIP 分支的 envs.SGLANG_OPT_USE_TOPK_V2 从 set(False) 改为 set(True)；SM120 分支保持不变，继续禁用 v2，形成跨平台差异的对照。
2. 平台编译隔离：在 python/sglang/kernels/jit/include/sgl_kernel/deepseek_v4/topk_impl.cuh 中，cooperative_groups.h 仅在非 ROCm 下引入；warp_inclusive_sum 在 HIP 下改用 kFullMask 并显式传 kWarpThreads 宽度；warp_sum_bool 针对 wave64 修复 ballot 结果与 __popc 位数不匹配的问题；TopKCluster 整体包进 #ifndef USE_ROCM，因为 CDNA 没有线程块集群与分布式共享内存。
3. dispatch 层同步条件化：在 python/sglang/kernels/jit/csrc/deepseek_v4/topk_v2.cuh 中，Cluster 类型别名、CLUSTER_TOPK_KERNEL 宏、topk_persistent_cluster_kernel、topk_small_batch_kernel 及 use_cluster 判断均以 USE_ROCM 隔离；两处 TensorMatcher 的设备选项从 kDLCUDA 改为 kDLGPU，使同一份 shape/strides 校验在 HIP 运行时也能命中设备。
4. 测试配套：在 test/registered/kernels/ops/attention/test_topk_v2.py 中新增 register_amd_ci(est_time=30, stage="jit-kernel-unit", runner_config="amd")，使既有约 278 个用例在 AMD CI 上执行；用例矩阵覆盖 trivial、Register2、Register4、Streaming、Cluster 各模板以及 8192/8193、16384/16385、65535/65536/65537、

batch 30/31、128/129 等边界。

5. 验证结果：GSM8k 1319 条准确率 0.945；核内基准长上下文加速 2.66-2.97 倍；E2E 长上下文（ISL/OSL 70k/200）TTFT 最大降约 576ms（cc=8）、ITL 最大约 -10%，但 TTT 增加约 4-7%。

关键文件：

- python/sglang/srt/server_args.py（模块 服务配置；类别 source；类型 core-logic；符号 _handle_model_specific_adjustments, SGLANG_OPT_USE_TOPK_V2）：平台开关的最终落点：把 HIP 分支的 SGLANG_OPT_USE_TOPK_V2 从 False 翻转为 True，是本次功能生效的唯一行为改动。
- python/sglang/kernels/jit/include/sgl_kernel/deepseek_v4/topk_impl.cuh（模块 内核适配；类别 source；类型 core-logic；符号 warp_inclusive_sum, warp_sum_bool, TopKCluster）：核心平台适配：条件编译 cooperative_groups、修正 HIP 的 shfl 与 ballot 语义、在 ROCm 下排除 TopKCluster；wave64 ballot 修复是本 PR 最有技术含量的部分。
- python/sglang/kernels/jit/csrc/deepseek_v4/topk_v2.cuh（模块 内核调度；类别 source；类型 core-logic；符号 TopKKernel, topk_persistent_cluster_kernel, topk_small_batch_kernel, CLUSTER_TOPK_KERNEL）：dispatch 层与模板匹配：Cluster 内核相关声明、宏与函数全部条件编译，同时把设备约束从 kDLCUDA 改为 kDLGPU 以适配 HIP。
- test/registered/kernels/ops/attention/test_topk_v2.py（模块 单元测试；类别 test；类型 test-coverage；符号 register_amd_ci）：把既有 v2 单元测试接入 AMD CI，防止平台适配回归；测试矩阵本身已覆盖各模板边界。

关键符号：_handle_model_specific_adjustments, warp_inclusive_sum, warp_sum_bool, TopKCluster, topk_persistent_cluster_kernel, topk_small_batch_kernel, TopKKernel, register_amd_ci

关键源码片段

python/sglang/srt/server_args.py

平台开关的最终落点：把 HIP 分支的 SGLANG_OPT_USE_TOPK_V2 从 False 翻转为 True，是本次功能生效的唯一行为改动。

```
# python/sglang/srt/server_args.py · _handle_model_specific_adjustments
# DeepSeek-V4 的模型级默认开关按平台分派：
# - SM120: 缺少 tcgen05/TMEM，继续禁用依赖 DeepGEMM 或大于 99KB SMEM 的 topk v2
# - HIP：本 PR 适配完成后，将 topk v2 从默认关闭改为默认开启
elif model_arch in ["DeepseekV4ForCausalLM"]:
    ...
    if is_sm120_supported():
        envs.SGLANG_OPT_FP8_WO_A_GEMM.set(False)
        envs.SGLANG_OPT_USE_TOPK_V2.set(False) # SM120 不满足 v2 的 SMEM 需求
        envs.SGLANG_OPT_USE_TILELANG_MHC_PRE.set(False)
        ...
    elif is_hip():
```

```
        envs.SGLANG_OPT_DEEPGEMM_HC_PRENORM.set(False)
        envs.SGLANG_OPT_FP8_WO_A_GEMM.set(False)
        envs.SGLANG_OPT_USE_JIT_INDEXER_METADATA.set(False)
        # 关键变更：此前 AMD 路径强制 v1（8 位粗粒度 key + 128KB LDS），
        # 现在默认走 v2（12 位 key + 寄存器驻留行），减少 refinement 候选数。
        # 若出现精度或性能回退，用户仍可通过 SGLANG_OPT_USE_TOPK_V2 环境变量回退。
        envs.SGLANG_OPT_USE_TOPK_V2.set(True)
        envs.SGLANG_OPT_USE_AITER_INDEXER.set(True)
        envs.SGLANG_OPT_USE_TILELANG_MHC_PRE.set(False)
        envs.SGLANG_OPT_USE_TILELANG_MHC_POST.set(False)
        envs.SGLANG_FP8_PAGED_MQA_LOGITS_TORCH.set(True)
        envs.SGLANG_OPT_USE_MULTI_STREAM_OVERLAP.set(False)
        envs.SGLANG_EAGER_INPUT_NO_COPY.set(True)
```

python/sglang/kernels/jit/include/sgl_kernel/deepseek_v4/topk_impl.cuh

核心平台适配：条件编译 cooperative_groups、修正 HIP 的 shfl 与 ballot 语义、在 ROCm 下排除 TopKCluster；wave64 ballot 修复是本 PR 最有技术含量的部分。

```
// python/sglang/kernels/jit/include/sgl_kernel/deepseek_v4/topk_impl.cuh
// 该文件同时被 CUDA 与 HIP 编译器使用，平台差异全部用 USE_ROCM 隔离。

#ifndef USE_ROCM
namespace cg = cooperative_groups; // 仅 CUDA 存在 cooperative_groups
#endif

// 逻辑 warp 内的 inclusive 扫描：HIP 的 __shfl_up_sync 需要显式传入宽度
//  （kWarpThreads），CUDA 版本则默认 32 线程即可。
SGL_DEVICE uint32_t warp_inclusive_sum(uint32_t lane_id, uint32_t val) {
#pragma unroll
    for (uint32_t offset = 1; offset < 32; offset *= 2) {
#ifndef USE_ROCM
        uint32_t n = __shfl_up_sync(0xFFFFFFFF, val, offset);
#else
        uint32_t n = __shfl_up_sync(kFullMask, val, offset, kWarpThreads);
#endif
        if (lane_id >= offset) val += n;
    }
    return val;
}

// 统计某个 predicate 在当前 warp 中的命中数。
// 注意：wave64 硬件上 __ballot 覆盖整个 wave（两个 32 线程逻辑 warp），
// 直接对结果做 __popc 会把 wave 低半段的统计同时报给两个逻辑 warp；
// 因此这里先根据 __lane_id() / kWarpSize 定位本 warp 所在的 32 位半区，
// 再把调用方 mask 移到对应半区，最后统计完整的 64 位。
SGL_DEVICE uint32_t warp_sum_bool(bool pred, uint32_t mask = 0xFFFFFFFF) {
#ifdef USE_ROCM
    const uint32_t half = __lane_id() / kWarpSize;
    return __popcll(__ballot(pred) & (static_cast<uint64_t>(mask) << (kWarpSize * half)));
```

```
#else
    return __popc(__ballot_sync(mask, pred));
#endif
}

// CDNA 架构没有线程块集群（thread-block cluster）与分布式共享内存，
// 因此超长序列（seq > 65536）使用的 TopKCluster 在 ROCm 下整体不参与编译，
// 由 Streaming 模板承接，避免每 CU 一个 block 的旧 v1 方案。
#ifndef USE_ROCM
template <uint32_t kClusterSize_>
struct TopKCluster : TopKRadixBase<10> {
    // ... 依赖 __cluster_dims__ 与分布式共享内存的实现
};
#endif
```

python/sglang/kernels/jit/csrc/deepseek_v4/topk_v2.cuh

dispatch 层与模板匹配：Cluster 内核相关声明、宏与函数全部条件编译，同时把设备约束从 kDLCUDA 改为 kDLGPU 以适配 HIP。

```
// python/sglang/kernels/jit/csrc/deepseek_v4/topk_v2.cuh（节选）
// 平台差异点集中在"集群内核"的声明与 dispatch 判断上：
// CUDA 的 thread-block cluster 需要 __cluster_dims__ 与分布式共享内存，
// CDNA（ROCm）没有等价能力，因此相关符号全部在 USE_ROCM 下剔除。

#ifndef USE_ROCM
using Cluster = impl::TopKCluster<8>; // 仅 CUDA 存在
constexpr uint32_t kClusterSize = Cluster::kClusterSize;
#endif

constexpr uint32_t kReg2MaxSeqLen = Register2::kMaxSeqLen; // 8192
constexpr uint32_t kReg4MaxSeqLen = Register4::kMaxSeqLen; // 16384
constexpr uint32_t kClusterFloor = 65536;
constexpr uint32_t kClusterMaxBatch = 512;

#define TOPK_KERNEL __global__ __launch_bounds__(kBlockSize, kOccupancy)
#ifndef USE_ROCM
#define CLUSTER_TOPK_KERNEL TOPK_KERNEL __cluster_dims__(1, kClusterSize, 1)
#endif

// plan 与 dispatch 的 TensorMatcher 设备约束统一改为 kDLGPU，
// 使同一份 shape/strides 校验在 HIP 运行时也能命中设备。
device_.set_options<kDLGPU>();

// plan 阶段根据 seq_len 分布决定 cluster 阈值；ROCm 下 use_cluster
// 恒不参与编译，超长行回退 Streaming 模板。
#ifndef USE_ROCM
const bool use_cluster = (max_seq_len > params.cluster_floor) &&
                         (batch_size <= kClusterMaxBatch);
#endif
```

评论区精华

该 PR 全程没有 review comment，唯一的正式审核是 HaiShaw 的 APPROVED，评语仅一句“HIP specific”，说明维护者确认这属于 HIP 平台特有改动、不影响其他平台分支。10 个提交的演进路径本身透露了主要的工程权衡：先移除 cooperative_groups.h（HIP 无此头文件），再修复 __shfl_up_sync 的 mask 与宽度、修复 __ballot 与 __popc 的 64 位位数不匹配，随后把 TopKCluster 与 cluster kernel 用 #ifndef USE_ROCM 注释掉，最终修正 gate 和 CI 注册。没有出现关于精度或性能取舍的争论，端到端 TTT 小幅上升也未在评审中被质疑。

- 无实质讨论，唯一审核为 HIP specific (design)：维护者确认改动为 HIP 平台特有，不影响其他平台分支；无未解决疑虑。

风险与影响

- 风险：
  - 默认行为变更：AMD DeepSeek-V4 部署默认从 v1 切到 v2，属于默认行为变化；E2E 的 TTT 反而增加 4-7%，对以总吞吐为目标的负载可能不划算，需通过环境变量 SGLANG_OPT_USE_TOPK_V2=false 显式回退。
  - wave64 ballot 逻辑复杂度：topk_impl.cuh 中 warp_sum_bool 的实现依赖 __lane_id() / kWarpSize 计算半区并统计全部 64 位。若在 wave32 模式或不同 HIP 版本下语义有差异，可能产生错误的候选统计并影响 topk 输出；现有测试只覆盖标准环境。
  - 集群路径缺失：CDNA 无线程块集群 /DSMEM，超长序列（seq > 65536）在 AMD 上只能走 Streaming 模板，无法获得 cluster kernel 的收益；测试矩阵中的 Cluster 用例在 AMD CI 上实际不会命中。
  - CI 状态未全绿：PR body 中 PR Test (Base) 与 AMD ROCm 7.2 两个槽位显示为失败（:x:），仅 PR Test (Extra) 通过，需确认是平台偶发还是 gate 改动引入。
  - 影响：影响范围限定在 AMD ROCm + DeepSeek-V4（dsv4 attention backend）组合：长上下文 prefill 的 TTFT 和 decode 的 ITL 有可感知改善（TTFT 最大约 -4%，ITL 最大约 -10%），但完整请求总时间 TTT 略增。对用户而言无需改启动参数即可获得新内核；对团队而言，后续维护 CUDA/HIP 共享内核时需持续维护 USE_ROCM 分支，避免集群路径行为漂移。CI 上新增一个 AMD jit-kernel-unit stage，增强平台回归覆盖。其他模型、SM120 和 NVIDIA 平台均不受影响。
  - 风险标记：AMD 默认行为变更，wave64 ballot 掩码复杂度，E2E TTT 略增，ROCm 无 cluster 路径，CI 槽位未全绿

关联脉络

- PR #36119 [AMD][DSV4] perf: MXFP8 MoRI dispatch to match the w4a8 MoE input format: 同为 AMD 上 DeepSeek-V4 的性能优化，涉及 token dispatcher 与 MoE runner 的平台适配，与本 PR 共享 DSV4 + AMD 技术线。
- PR #36130 [AMD][DSV4] perf: bound the MoRI receive buffer during decode: 同为 AMD DeepSeek-V4 decode 阶段性能优化，与本 PR 一样通过平台化改动提升长上下文场景吞吐。

- PR #36547 Fix DeepSeek V4 multistream QKV buffer lifetime: DeepSeek-V4 模型路径的修复，与本 PR 同属 deepseek_v4 内核与运行时稳定性迭代。
- PR #36356 [AMD] Enable aiter mla asm path through padding attn heads for Kimi K3: 同为 AMD 平台启用新内核路径（aiter MLA），与本 PR 属于同一类 ROCm 平台能力补齐。