PR #25898 完整报告

sgl-project/sglang

[AMD] Dsv4/pr1 fix run time issue

合并时间：2026-05-24 07:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25898

执行摘要

- 一句话：修复 DSV4 在 AMD GPU 上的运行时崩溃与 JIT 不兼容
- 推荐动作：推荐在 AMD 平台上部署 DeepSeek-V4 的团队仔细审查此 PR，特别是 JIT 内核的 HIP 兼容细节和 Triton fallback 的选择。对于未使用 AMD 的团队，可忽略此 PR。关于 rounding 不一致的问题建议与作者确认后续修复。

功能与动机

根据 PR body，DSV4 在 AMD GPU 上遇到三个运行时失败：

1. GPU fault (HSA_STATUS_ERROR_EXCEPTION) — CompressStatePool 初始化缺少 swa_page_size，默认为 0，导致 translate_from_swa_loc_to_state_loc() 中除零，产生越界索引触发硬件异常。
2. Model registration failure — fused_softmax_pool_triton 在 compress_hip.py 中被模块级导入但不存在于 deepseek_v4_rope.py 中，阻塞模型类注册。
3. JIT kernel incompatibilities — Wave64 shuffle intrinsics、DLPack 设备类型、FP8 转换和 kernel launch API 在 CUDA 和 HIP 之间不同。

实现拆解

1. JIT kernel HIP 兼容基础设施：修改共享头文件（warp.cuh、tensor.h、utils.cuh、fp8_utils.cuh、c128_v2.cuh、c4_v2.cuh、utils.py），为 wave64 架构添加 HIP 原生指令适配、设备类型抽象、跳过不支持的架构断言、软件 FP8 转换函数。
2. 运行时修复端口：修改 deepseek_v4.py（修复 compress_hip.py 导入缺失符号问题，添加 swa_page_size 到 CompressStatePool 初始化，引入 cos_cache/sin_cache 缓存等）、deepseek_v2.py（环境变量和辅助函数）、compress_hip.py（添加 fallback 机制和 fused_softmax_pool_triton 防御性导入）。
3. Triton fallback 实现：新增 triton/hash_topk.py（用于 hash_topk 的 Triton 版本）、triton_store_cache.py（包含 fused store flashmla 和 indexer 的 Triton 实现）、compress_hip.py 中的 fused_compress_triton 选项。
4. AOT 编译的 sgl-kernel 算子：在 elementwise.py 中添加 dsv4_fused_q_norm_rope、dsv4_fused_k_norm_rope_flashmla、dsv4_fused_q_indexer_rope_hadamard_quant Python 包装，并添加相应的 .cu/.hip 源文件、sgl_kernel_ops.h 声明以及构建系统集成。

5. 配置与测试配套：添加环境变量控制（如 SGLANG_USE_AITER、SGLANG_OPT_USE_FUSED_QK_NORM_ROPE 等），并在 PR body 中提供了 AMD 和 B200 的 GSM8K 准确率测试结果（AMD 94.0%，B200 95.4%）。

关键文件：

- python/sglang/srt/layers/fused_qk_norm_rope_store.py（模块 融合内核；类别 source；类型 core-logic；符号 _batched_rmsnorm, _gptj_rotate, _batched_rope, _fused_qk_norm_rope_store_kernel）：新增核心融合 Triton 内核，将 Q per-head RMSNorm + KV RMSNorm + RoPE + FP8 量化 + paged SWA store 合并为一个 kernel，是 AMD 路径的关键性能优化。
- python/sglang/jit_kernel/triton_store_cache.py（模块 JIT 内核；类别 source；类型 core-logic；符号 _triton_fused_store_flashmla_kernel, triton_fused_store_flashmla, _triton_fused_store_indexer_kernel, triton_fused_store_indexer）：新增 Triton fused store cache 内核，支持 FlashMLA 和 indexer 路径的 FP8 量化与 paged scatter，是 AMD fallback 的核心组件。
- python/sglang/jit_kernel/triton/hash_topk.py（模块 JIT 内核；类别 source；类型 core-logic；符号 _hash_topk_triton_kernel, hash_topk_triton）：新增 Triton 实现的 hash_topk fallback，CUDA 原生 kernel 使用 CUDA-only 原语，此 Triton 版本使其可在 ROCm 上运行。
- python/sglang/srt/models/deepseek_v4.py（模块 模型核心；类别 source；类型 data-contract；符号 _fused_rmsnorm_fp8_quant）：核心模型文件，修复了运行时问题，包括添加 cos_cache/sin_cache 缓存、fused_rmsnorm_fp8_quant 函数、SGLANG_USE_AITER 支持等。
- sgl-kernel/python/sgl_kernel/elementwise.py（模块 元素算子；类别 source；类型 core-logic；符号 dsv4_fused_q_norm_rope, dsv4_fused_k_norm_rope_flashmla, dsv4_fused_q_indexer_rope_hadamard_quant）：添加 DSV4 融合 kernel 的 Python 包装（dsv4_fused_q_norm_rope、dsv4_fused_k_norm_rope_flashmla、dsv4_fused_q_indexer_rope_hadamard_quant），连接到 AOT 编译的 sgl-kernel 实现。
- python/sglang/srt/layers/attention/dsv4/compress_hip.py（模块 注意力压缩；类别 source；类型 dependency-wiring；符号 use_fused_compress_triton, _get_freqs_cis_real）：修复了 fused_softmax_pool_triton 导入失败问题，添加了 Triton fallback 融合压缩选项和频率缓存。

关键符号：_batched_rmsnorm, _gptj_rotate, _batched_rope, _fused_qk_norm_rope_store_kernel, fused_qk_norm_rope_swa_store, _triton_fused_store_flashmla_kernel, triton_fused_store_flashmla, _triton_fused_store_indexer_kernel, triton_fused_store_indexer, triton_fused_store_cache, _hash_topk_triton_kernel, hash_topk_triton, _fused_rmsnorm_fp8_quant, dsv4_fused_q_norm_rope, dsv4_fused_k_norm_rope_flashmla, dsv4_fused_q_indexer_rope_hadamard_quant, use_fused_compress_triton, _get_freqs_cis_real

评论区精华

gemini-code-assist[bot] 在 review 中指出 fp8_utils.cuh 中的 round-half-up 与 sgl-kernel CUDA 实现中的 round-to-nearest-even 不一致，可能导致精度差异，建议统一为 round-to-nearest-even。该问题尚未解决，历史记录中未发现后续改动。

- 软件 FP8 转换 rounding 模式不一致 (correctness): 未解决，review 中未看到作者回复或后续修改。

风险与影响

- 风险：
  1. 精度风险：新引入的软件 FP8 转换函数在 ROCm 上使用 round-half-up，与 CUDA 的 round-to-nearest-even 不一致，可能导致跨平台精度差异（由 review 指出）。
  2. 性能风险：Triton fallback 实现（如 hash_topk、fused_store_cache）在 AMD 上可能比原生 CUDA 内核慢，但当前仅用于 HIP 路径。
  3. 兼容性风险：新增的环境变量和条件导入（如 SGLANG_USE_AITER）需要用户正确配置，否则可能不会 fallback 到正确实现。
  4. 测试覆盖：本 PR 没有直接添加单元测试，依赖端到端准确率测试确认正确性。 - 影响：用户：AMD GPU 用户现在可以在 MI300X/MI350X 上运行 DSV4 推理，准确率与 B200 基线一致（94.0% vs 95.4%）。系统：新增了多个 JIT 内核和 sgl-kernel 算子，增加了二进制大小和首次 JIT 编译时间。团队：AMD 团队需要维护这些 HIP 专用代码，但设计上尽量通过条件编译和 fallback 最小化重复。 - 风险标记：rounding precision inconsistency, 缺少测试覆盖 , 新增 fallback 路径性能退化

关联脉络

- 暂无明显关联 PR