PR #28722 完整报告

sgl-project/sglang

[AMD] Optimize o_proj gemm and attn output rope performance

合并时间：2026-06-20 17:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28722

执行摘要

- 一句话：优化 DeepSeek-V4 在 AMD GPU 上的 FP8 GEMM 和 RoPE 性能
- 推荐动作：
  1. 建议在后续 PR 中修复类型转换问题，对 out_real 和 out_imag 添加 .to(x_ptr.dtype.element_ty)。
  2. 使用 get_bool_env_var 初始化模块级开关，使环境变量覆盖生效。
  3. 补充单元测试覆盖新内核的正确性（如与旧内核输出一致性测试）。
  4. 关注 #28783 中的 BLOCK_M 调优结果，适时合并。
  5. 值得学习的点：通过松耦合的模块级标志和模型构造函数 opt-in，在不影响其他模型的前提下对特定模型实施性能优化。

功能与动机

DeepSeek-V4 在 AMD GPU 上的 MLA 投影 w8a8 GEMM 使用了 Triton 内核，但 CK bpreshuffle 内核在目标形状下更快；注意力输出逆 RoPE 在 HIP 上回退到每 token 一个程序的 Triton 内核，启动开销大且访问模式为非合并的跳步加载。PR 旨在通过切换内核选择策略和设计批量 / 连续加载内核来消除这两处性能瓶颈。

实现拆解

1. fp8_utils.py：添加模块级布尔开关 _FORCE_CK_W8A8 及其设置函数 set_force_ck_w8a8()，在 use_aiter_triton_gemm_w8a8_tuned_gfx950() 中优先检查该开关：若为 True 则返回 False，使调用方绕过 Triton 内核选择 CK bpreshuffle 内核。
2. deepseek_v4_rope.py：新增两个 Triton JIT 内核：apply_rotary_emb_triton_kernel_batched（每程序处理 BLOCK_M 个 token，减少启动次数）和 apply_rotary_emb_contig_kernel（连续加载 rope 切片，实现合并访问）。添加模块级开关 _USE_BATCHED_ROPE 及 set_batched_rope()，在 apply_rotary_emb_triton() 入口处根据开关选择新内核（3D 逆 RoPE 时选 contiguous 内核，其余选 batched 内核）或回退旧行为。
3. deepseek_v4.py：在 DeepseekV4ForCausalLM.__init__() 中，当 _is_hip 为真时调用 set_force_ck_w8a8(True) 和 set_batched_rope(True)，使 DeepSeek-V4 默认启用两项优化；其他模型因开关默认关闭不受影响。

4. 环境变量覆盖：代码注释声明可通过 SGLANG_FORCE_CK_W8A8 和 SGLANG_ROPE_BATCHED 环境变量覆盖，但当前实现未实际读取这些变量（review 中已指出，留待后续 PR 修复）。

关键文件：

- python/sglang/srt/layers/deepseek_v4_rope.py（模块 旋转位置编码；类别 source；类型 core-logic；符号 apply_rotary_emb_triton_kernel_batched, apply_rotary_emb_contig_kernel, set_batched_rope）：新增两个 Triton JIT 内核（batched 和 contiguous）以及模块级开关 set_batched_rope，是注意力输出逆 RoPE 优化的核心。
- python/sglang/srt/layers/quantization/fp8_utils.py（模块 量化路径；类别 source；类型 core-logic；符号 set_force_ck_w8a8）：新增 _FORCE_CK_W8A8 开关和 set_force_ck_w8a8，控制 MLA 投影 GEMM 的内核选择。
- python/sglang/srt/models/deepseek_v4.py（模块 模型入口；类别 source；类型 data-contract）：模型构造函数中通过 opt-in 方式启用两项优化，且仅对 HIP 生效，其他模型不受影响。

关键符号：set_force_ck_w8a8, set_batched_rope, apply_rotary_emb_triton_kernel_batched, apply_rotary_emb_contig_kernel

关键源码片段

python/sglang/srt/layers/quantization/fp8_utils.py

新增 _FORCE_CK_W8A8 开关和 set_force_ck_w8a8，控制 MLA 投影 GEMM 的内核选择。

```
# _FORCE_CK_W8A8 是一个模块级开关，用于强制 MLA 投影的 w8a8-block GEMM 使用
# CK bpreshuffle 内核（而不是 Triton 内核）。默认关闭；DeepSeek-V4 在初始化时
# 调用 set_force_ck_w8a8(True) 开启。环境变量 SGLANG_FORCE_CK_W8A8=1 也可作为覆盖。
_FORCE_CK_W8A8: bool = False

def set_force_ck_w8a8(enabled: bool = True) -> None:
    """设置强制使用 CK bpreshuffle 内核的标志"""
    global _FORCE_CK_W8A8
    _FORCE_CK_W8A8 = enabled

def use_aiter_triton_gemm_w8a8_tuned_gfx950(n: int, k: int) -> bool:
    """判断指定 (n, k) 形状是否应使用 Triton 内核。
    当 _FORCE_CK_W8A8 为 True 时，直接返回 False，表示不使用 Triton，
    从而路由到 CK bpreshuffle 内核。
    """
    if _FORCE_CK_W8A8:
        return False
    # 已调优的形状列表，使用 Triton 内核
    return (n, k) in [
        (1024, 8192),
        (16384, 1536),
        (2112, 7168),
```

```
        (3072, 1536),
        (32768, 8192),
        (4096, 7168),
        (4608, 7168),
        (512, 7168),
        (7168, 2048),
        (7168, 2304),
        (7168, 16384),
        (7168, 256),
        (8192, 1024),
        (8192, 32768),
    ]
```

评论区精华

Review 中主要讨论了四个问题：

- 类型转换：*gemini-code-assist* 指出 batched 内核中存储输出前应显式转换元素类型，作者未处理。
- 环境变量未生效：两个内核的环境变量覆盖均未实际实现，HaiShaw 要求后续 PR 修复。
- BLOCK_M 最优性：HaiShaw 质疑 32 是否对 gfx950/gfx942 最优，作者回应已创建 #28783 进行调优。
- gfx950/gfx942 粒度：HaiShaw 询问是否需要区分，作者确认已在两平台测试，CK 内核均更快，无需控制。
- batched kernel 中输出类型未转换 (correctness): 作者未在 PR 中处理；HaiShaw 要求后续 PR 处理。
- 环境变量 SGLANG_ROPE_BATCHED 未实际生效 (design): 作者未处理；HaiShaw 要求后续 PR 处理。
- 环境变量 SGLANG_FORCE_CK_W8A8 未实际生效 (design): 同前，未处理。
- BLOCK_M 参数对 gfx950 和 gfx942 的最优性 (performance): 作者创建后续 PR 进行调优，本次 PR 保持默认。
- 是否需要区分 gfx950 和 gfx942 的粒度控制 (design): 作者确认无需区分，讨论关闭。

风险与影响

- 风险：
  1. 类型转换风险：在 apply_rotary_emb_triton_kernel_batched 中，输出 out_real 和 out_imag 未显式转换为 x_ptr 的元素类型，可能在 ROCm/HIP 上导致编译警告或运行时类型不匹配错误。
  2. 环境变量覆盖未实现：代码注释声称的环境变量 SGLANG_FORCE_CK_W8A8 和 SGLANG_ROPE_BATCHED 目前不能实际生效，可能误导依赖该功能的用户。
  3. 缺少测试覆盖：新内核无单元测试，仅依赖端到端基准测试，可能遗漏边界情况（如空 tensor、极端序列长度）。

4. 仅 ROCm 平台启用：优化仅针对 HIP 环境，CUDA 上行为不变，但未验证 CUDA 下开关关闭是否引入回归（理论上无影响）。 - 影响：对 DeepSeek-V4 用户在 AMD ROCm 平台（gfx950/gfx942）带来显著性能提升：纯预填阶段约 8-9%，端到端约 3-5%。其他模型不受影响。团队需在后续 PR #28783 中完成 BLOCK_M 调优，并修复环境变量实现。长期来看，此 PR 确立了一种通过模块级开关和模型级 opt-in 进行性能切换的通用模式，便于未来为其他模型启用类似优化。 - 风险标记：类型转换潜在编译警告，环境变量覆盖未实现，缺少测试覆盖

关联脉络

- PR #28783 后续 BLOCK_M 调优 PR: 作者在 review 中提及已创建该 PR 来优化 BLOCK_M 配置，后续需关注。