

PR #36379 完整报告

sgl-project/sglang

fix(lora): build the MoE LoRA align JIT kernel on ROCm

合并时间: 2026-08-28 16:18

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36379>

执行摘要

- 一句话: 修复 MoE LoRA align 内核在 ROCm 上的编译失败
- 推荐动作: 建议精读的重点:
- 学习如何在 JIT 内核中做平台适配, 尤其是 hipify 缺失时的处理方法。
- 理解依赖闭包与缓存失效的关系, 避免滥用公共头文件。
- 参考测试注册模式, 为平台相关内核补充对应 CI。

功能与动机

MoE LoRA align JIT 内核在 ROCm 上无法编译, 因为 JIT 构建不执行 hipify, 导致 `cub/cub.cuh`、`cudaDevAttrMaxSharedMemoryPerBlockOptin` 等 CUDA API 在 HIP 环境下不存在。该内核由 `_compute_lora_alignment` 调用, 导致在 ROCm 上 serve MoE 模型加 LoRA adapter 时首次前向即中止。

实现拆解

实现拆解:

1. 修改 `moe_lora_align_kernel.cu` (核心源码):
 - 在 `USE_ROCM` 宏下引入 `hipcub/hipcub.hpp`, 并设置 `namespace cub = hipcub` 别名。
 - 将三个 CUDA API 宏映射到 HIP 拼写: `cudaDevAttrMaxSharedMemoryPerBlockOptin` → `hipDeviceAttributeSharedMemPerBlockOptin`, `cudaFuncSetAttribute` → `hipFuncSetAttribute`, `cudaFuncAttributeMaxDynamicSharedMemorySize` → `hipFuncAttributeMaxDynamicSharedMemorySize`。
 - 修改调用点: `cudaFuncSetAttribute` 通过 `std::bit_cast<const void*>` 传入函数指针, 以匹配 `hipFuncSetAttribute` 的 `const void*` 签名。
 - 这些别名刻意不加在 `utils.cuh` 中, 以避免大量 JIT 缓存失效导致 CI 超时。
2. 修改测试文件 `test_moe_lora_align_block_size.py`:
 - 导入 `register_amd_ci`, 并注册 AMD CI 阶段, 使该测试在 AMD CI 上运行, 确保内核在 ROCm 上被编译。
3. 配套说明:
 - 无其他配置或部署改动。

关键文件:

- python/sglang/kernels/jit/csrc/lora/moe_lora_align_kernel.cu (模块 内核; 类别 source; 类型 core-logic; 符号 setup_kernel_smem_once, MoeLoraAlignBlockSizeKernel::forward) : 核心修复文件, 添加 ROCm 平台适配, 修复 JIT 内核编译失败。
- test/registered/kernels/ops/moe/test_moe_lora_align_block_size.py (模块 测试; 类别 test; 类型 test-coverage) : 新增 AMD CI 注册, 确保该测试在 ROCm 上运行, 防止回归。

关键符号: setup_kernel_smem_once, MoeLoraAlignBlockSizeKernel::forward

关键源码片段

python/sglang/kernels/jit/csrc/lora/moe_lora_align_kernel.cu

核心修复文件, 添加 ROCm 平台适配, 修复 JIT 内核编译失败。

```
// moe_lora_align_kernel.cu 关键片段
#ifdef USE_ROCM
// JIT 构建不执行 hipify, 这里手动映射 CUDA 拼写到 HIP。
// 刻意只在本地定义, 不放入 utils.cuh:
// utils.cuh 被几乎所有 JIT 内核间接依赖, 改动它会使整个 JIT 缓存失效。
#include <hipcub/hipcub.hpp>
#include <bit>
namespace cub = hipcub;
#define cudaDevAttrMaxSharedMemoryPerBlockOptin
hipDeviceAttributeSharedMemPerBlockOptin
#define cudaFuncSetAttribute hipFuncSetAttribute
#define cudaFuncAttributeMaxDynamicSharedMemorySize
hipFuncAttributeMaxDynamicSharedMemorySize
#endif

// 调用处: 适配 hipFuncSetAttribute 需要 const void* 而非函数指针
const auto fptr = std::bit_cast<const void*>(kernel);
RuntimeDeviceCheck(cudaFuncSetAttribute(fptr,
    cudaFuncAttributeMaxDynamicSharedMemorySize, shared_mem));
```

评论区精华

Review 讨论的核心:

- HaiShaw 询问 **WARP_SIZE** 在 gfx942/gfx950/gfx1250 上的取值。Arist12 回应: **WARP_SIZE** 是逻辑 tile 宽度, 而非硬件 wavefront; 该内核不含 wave primitive, 布局与 wave 宽度无关, 并用 **WARP_SIZE 64** 验证结果一致。
- 关于别名位置的权衡: 作者解释在 **utils.cuh** 中添加别名会导致全部 JIT 缓存失效, 引发 CI 超时, 因此刻意限制在局部文件。
- **WARP_SIZE** 与硬件 wavefront 的关系 (question): Arist12 解释 **WARP_SIZE** 是逻辑 tile 宽度, 不含 wave primitive, 布局不依赖 wave 宽度, 且用 **WARP_SIZE 64** 验证结果一致。

风险与影响

- 风险：风险点：
- 该改动仅影响 ROCm 下 JIT 内核的编译，CUDA 路径不受影响（`#ifndef USE_ROCM` 分支）。
- `std::bit_cast` 需要 C++20，若编译环境不支持可能编译失败，但根据作者验证在 ROCm 7.2 下通过。
- 测试覆盖新增了 AMD CI 注册，但 `est_time=30` 可能接近 CI 超时，存在 CI 不稳定风险。
- 修改依赖引入（hipCUB）可能影响其他 ROCm 内核编译，但影响范围限于该文件。
- 影响：影响范围：
- 对用户：ROCm 平台上 MoE 模型 + LoRA adapter 的推理不再崩溃，可正常 serving。
- 对系统：新增 AMD CI 测试覆盖，确保该内核在 ROCm 上持续可编译。
- 对团队：需要维护 hipCUB 相关依赖，但改动局部化风险低。
- 风险标记：平台适配，JIT 缓存失效风险，新增 CI 覆盖

关联脉络

- PR #36759 bugfix for `index_fill_` on NPU: 同为平台相关内核 bugfix，涉及内存分配器中的 `index_fill_` 性能问题，虽非 JIT 但指示平台适配类 PR 的模式。
- PR #35613 [diffusion] refactor: scope model-specific API parameters: 同为模型 / 内核相关的重构与平台适配，虽不直接相关，但反映仓库在跨硬件适配上的持续投入。