PR #2274 完整报告

THUDM/slime

[ROCm] Support the INT4 QAT kernel on ROCm

合并时间：2026-08-16 17:44

原文链接：http://prhub.com.cn/THUDM/slime/pull/2274

执行摘要

- 一句话：为 ROCm 构建 INT4 QAT 内核并修复 hipcc 兼容性
- 推荐动作：值得精读。该 PR 展示了如何通过条件编译适配不同 GPU 编译器（nvcc vs hipcc），但改动量较小。关注点在于 ROCm 构建配置的通用性和未来维护成本。

功能与动机

PR 描述指出 int4_qat 内核在 ROCm PyTorch 安装下无法构建，导致 INT4 QAT 路径在 AMD GPU 上不可用。需要修复两个阻塞点：HIP warp-shuffle 掩码和 hipcc 下的 const_data_ptr 符号 mangling 不匹配。

实现拆解

1. 修改 fake_int4_quant_cuda.cu：为核心内核函数 launch_int4_quant_kernel 添加 HIP 平台条件编译，将 __shfl_xor_sync 替换为无掩码的 __shfl_xor（适用于 32 线程组），并处理 const_data_ptr 模板在 hipcc 下的链接问题。
2. 修改 setup.py：检测 ROCm 环境（torch.version.hip），设置 PYTORCH_ROCM_ARCH 默认 gfx950，并在非 ROCm 时仅传递 nvcc 专用标志（如 -gencode、--expt-relaxed-constexpr），避免 hipcc 拒绝。
3. 验证：在 MI355X / ROCm 7.2 上验证 144 个 case 与 CUDA 逐位一致，并通过 E2E 转换工具验证 Qwen3-0.6B 的 INT4 转换质量。

关键文件：

- slime/backends/megatron_utils/kernels/int4_qat/setup.py（模块 后端；类别 source；类型 core-logic）：核心构建配置变更，区分 CUDA 和 ROCm 编译参数，直接影响内核能否构建成功。
- slime/backends/megatron_utils/kernels/int4_qat/fake_int4_quant_cuda.cu（模块 后端；类别 source；类型 core-logic；符号 warpReduceMax, warpReduceMin, fake_int4_quant_cuda）：内核源码的 HIP 兼容性修复，是功能实现的直接载体。

关键符号：warpReduceMax, warpReduceMin, fake_int4_quant_cuda

关键源码片段

slime/backends/megatron_utils/kernels/int4_qat/fake_int4_quant_cuda.cu

内核源码的 HIP 兼容性修复，是功能实现的直接载体。

```
// slime/backends/megatron_utils/kernels/int4_qat/fake_int4_quant_cuda.cu
#include <torch/extension.h>
#include <ATen/cuda/CUDAContext.h>

// HIP 的 __shfl_xor_sync 要求 64 位掩码（wavefront 为 64 条 lane），0xFFFFFFFF 无法编译。
// 归约只在 32 条 lane 的组内进行，因此使用无掩码的 __shfl_xor 并指定宽度 32 即可。
#if defined(__HIP_PLATFORM_AMD__)
#define WARP_XOR(val, mask) __shfl_xor((val), (mask), 32)
#else
#define FINAL_MASK 0xFFFFFFFF
#define WARP_XOR(val, mask) __shfl_xor_sync(FINAL_MASK, (val), (mask), 32)
#endif

// 设备端归约函数：使用 WARP_XOR 替代 __shfl_xor_sync（原代码中 mask 从 16 递减到 1）
__device__ __forceinline__ float warpReduceMax(float val) {
    #pragma unroll
    for (int mask = 16; mask > 0; mask >>= 1)
        val = fmaxf(val, WARP_XOR(val, mask));
    return val;
}

__device__ __forceinline__ float warpReduceMin(float val) {
    #pragma unroll
    for (int mask = 16; mask > 0; mask >>= 1)
        val = fminf(val, WARP_XOR(val, mask));
    return val;
}

// 在量化内核启动处，针对 HIP 平台使用静态转换以解决 clang 与 gcc 的模板 mangling 差异
void fake_int4_quant_cuda(...) {
    ...
    launch_int4_quant_kernel<scalar_t>(
#if defined(__HIP_PLATFORM_AMD__)
        // templated const_data_ptr<T> 在 hipcc 下无法链接：clang 对 enable_if 的 mangling
        与构建 libtorch 的 gcc 不同
        static_cast<const scalar_t*>(x.const_data_ptr()),
#else
        x.const_data_ptr<scalar_t>(),
#endif
        out.data_ptr<scalar_t>(),
        out_scale.data_ptr<scalar_t>(),
        out_zero.data_ptr<scalar_t>(),
        ...);
}
```

评论区精华

无 review 评论，因此没有公开的讨论记录。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. CUDA 路径风险低：所有改动均在 #if defined(__HIP_PLATFORM_AMD__) 或 torch.version.hip is not None 保护内，确保 CUDA 编译路径不变。
  2. ROCm 路径风险：依赖 PYTORCH_ROCM_ARCH 环境变量，默认 gfx950 可能不适用于所有 AMD GPU，用户需显式设置。
  3. API 兼容性：const_data_ptr 的静态转换方式可能影响未覆盖的调用点，需确认 fake_int4_quant_cuda 的所有调用路径。 - 影响：影响范围：直接影响 INT4 QAT 内核的构建和运行，主要涉及 ROCm 用户（AMD GPU 集群）。对 CUDA 用户无影响。团队影响：扩大了 Slime 在 AMD 硬件上的可用性，可能需要更新文档或 CI 以覆盖 ROCm 构建测试。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #2271 fix transform_ue8m0 in fp8 convert: 同为量化转换相关改动，涉及 Megatron 到 HF 的转换逻辑，可能共享类似内核或转换工具。
- PR #2267 Fix model convert when use latest megatron: 与模型转换工具相关，本 PR 的 E2E 验证使用了 convert_hf_to_int4_direct.py，可能与这些工具联动。