PR #40428 完整报告

vllm-project/vllm

[Bugfix][CPU][RISC-V] Clamp exp() input to prevent NaN

合并时间：2026-04-22 17:38

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40428

执行摘要

本 PR 修复了 RISC-V CPU 平台上向量指数函数 *exp()* 因未钳制输入导致 NaN 的数值 bug。在 FP32Vec8 和 FP32Vec16 的 *exp()* 方法中添加了与 x86 AVX-512 和 ARM NEON 一致的输入钳制（[-87.33, 88.72]），确保极值输入时多项式评估不产生 -inf × 0.0 = NaN。同时移除了 BF16 条件编译，简化代码。该修复使 RISC-V 注意力路径中的 softmax 计算恢复正常。

功能与动机

调试 RISC-V 注意力路径时发现 softmax 产生 NaN，最终定位到 FP32Vec16::exp() 函数。当输入值极大（例如 softmax 前 logits 很大）时，多项式评估后的缩放因子可能导致 -inf * 0.0，根据 IEEE 754 此运算结果为 NaN。PR 描述明确指出此问题，并选择与 x86/ARM 平台相同的钳制边界，保证跨平台行为一致。

实现拆解

1. 简化 BF16 宏定义

文件 csrc/cpu/cpu_types_riscv_impl.hpp 开头，移除了 #ifdef RISCV_BF16_SUPPORT 条件编译，始终包含 BFloat16 支持。因为 RISC-V 平台要么原生支持 BF16，要么通过 FP32 模拟回退支持，所以无需条件分支。

2. FP32Vec8::exp() 添加输入钳制

在多项式评估开始前，使用 RVV 向量内建函数对输入 reg 进行双向钳制：

随后所有多项式计算使用钳制后的 x，而非原始 reg。

3. FP32Vec16::exp() 添加相同钳制

完全相同的逻辑，仅 LMUL 从 256 改为 512。

4. 无测试文件变更

此 PR 未包含测试用例，但改动逻辑简单，且已在 RISC-V 实际环境中验证。

csrc/cpu/cpu_types_riscv_impl.hpp

包含所有核心改动：移除 BF16 条件编译、FP32Vec8 和 FP32Vec16 的 *exp()* 方法添加输入钳制。

关键源码片段

csrc/cpu/cpu_types_riscv_impl.hpp

包含所有核心改动：移除 BF16 条件编译、FP32Vec8 和 FP32Vec16 的 exp() 方法添加输入钳制。

```
// FP32Vec8::exp() 的变更后实现（位于 csrc/cpu/cpu_types_riscv_impl.hpp）
FP32Vec8 exp() const {
    // 钳制输入以防止 NaN：exp(-inf) 必须返回 0，而非 NaN。
    // 若不钳制，多项式最后一步的 -inf * 0.0 会产生 NaN。
    // 此策略与 x86 AVX-512 和 ARM NEON 一致。
    constexpr float exp_lo = -87.3365447505f; // ln(FLT_MIN)
    constexpr float exp_hi = 88.7228391117f; // ln(FLT_MAX)
    // 先取 max(reg, exp_lo) 再取 min(..., exp_hi) 实现双向钳制
    fixed_fp32x8_t x = RVVI(__riscv_vfmin_vf_f32, LMUL_256)(
        RVVI(__riscv_vfmax_vf_f32, LMUL_256)(reg, exp_lo, VEC_ELEM_NUM), exp_hi,
        VEC_ELEM_NUM);
    // 后续多项式评估使用钳制后的 x 而非原始 reg
    const float inv_ln2 = 1.44269504088896341f;
    fixed_fp32x8_t x_scaled =
        RVVI(__riscv_vfmul_vf_f32, LMUL_256)(x, inv_ln2, VEC_ELEM_NUM);
    // ... 其余多项式代码不变
}
```

评论区精华

> gemini-code-assist[bot] 评论：下界 -87.33（ln(FLT_MIN)）会导致 exp(-inf) 返回 FLT_MIN 而非严格 0.0f；若需严格零输出，应使用约 -103.0f。但评论者也指出，PR 描述中明确提到匹配 x86/ARM 策略，当前设计是预期行为。

结论：作者和合并者接受当前边界，未修改。

风险与影响

- 数值边界精度：exp(-inf) 返回 FLT_MIN（~1.18e-38）而非 0，在 softmax 归一化中可能产生微小正概率，但行为与主流平台一致，风险低。
- 回归风险：仅修改 RISC-V 特定路径，不影响其他平台。改动逻辑简单，回归风险低。
- 性能影响：增加两次向量比较操作，可忽略。
- 影响范围：仅 RISC-V CPU 用户，修复了注意力路径的 NaN 问题，使 RISC-V 推理更稳定。

关联脉络

本 PR 是 RISC-V 平台支持系列中的数值稳定性修复。类似地，PR #40176 修复了 ROCm 平台的注意力数值问题，体现了跨平台数值一致性的持续关注。此前还有多个与 CPU/RISC-V 相关的 PR（如 #39835 等），表明 vLLM 对 RISC-V 架构的投入。