# PR #48868 完整报告

vllm-project/vllm

[Helion] Fix degenerate scale_ub in kernel input generators

合并时间：2026-07-17 03:57

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48868

## 执行摘要

- 一句话：修复 Helion kernel 测试输入中 scale_ub 退化问题
- 推荐动作：建议合并。这是一次小但有意义的测试修复，消除了测试中的 false positive，使后续 Helion kernel 的测试更加可靠。

## 功能与动机

原本 scale_ub = torch.mean(input) 对于零均值的 torch.randn 输入，mean(input) ≈ 0，导致所有量化尺度被钳位到 min_scaling_factor，输出完全饱和。这样 autotuner 精度检查（以及任何使用该输入生成器的正确性测试）比较两个饱和张量，实际上是无效的。

## 实现拆解

1. 分析问题：识别出 generate_inputs() 中 scale_ub = torch.mean(input) 对于零均值输入导致严重饱和。
2. 设计修复策略：为每个操作数计算被钳位量的幅度（绝对值），取均值与最大值的平均值作为 scale_ub，使得钳位部分激活。
3. 应用到四个文件：修改 vllm/kernels/helion/ops/ 下的四个文件：
   - silu_and_mul_per_block_quant.py：计算激活的绝对值，使用 SiluAndMul.forward_native。
   - dynamic_per_token_scaled_fp8_quant.py：计算 linputl。
   - rms_norm_per_block_quant.py：计算 RMS 归一化后加权输出的绝对值。
   - rms_norm_dynamic_per_token_quant.py：同上。
4. 验证：在 dynamic_per_token_scaled_fp8_quant 上测试，饱和比例从 99.9% 降低到 ~5%（正常水平）。

关键文件：

- vllm/kernels/helion/ops/dynamic_per_token_scaled_fp8_quant.py（模块 Kernel 测试；类别 test；类型 test-fix）：修复了 generate_inputs() 中 scale_ub 的计算，使用 linputl 的均值与最大值的一半。
- vllm/kernels/helion/ops/rms_norm_dynamic_per_token_quant.py（模块 Kernel 测试；类别 test；类型 test-fix）：修复了 generate_inputs() 中 scale_ub 的计算，使用 RMS 归一化后加权输出的幅度均值与最大值的一半。

- vllm/kernels/helion/ops/rms_norm_per_block_quant.py（模块 Kernel 测试；类别 test；类型 test-fix）：修复了 generate_inputs() 中 scale_ub 的计算，与 rms_norm_dynamic_per_token_quant.py 类似但针对 per-block 量化。
- vllm/kernels/helion/ops/silu_and_mul_per_block_quant.py（模块 Kernel 测试；类别 test；类型 test-fix）：修复了 generate_inputs() 中 scale_ub 的计算，使用 SiLU-and-mul 激活的幅度均值与最大值的一半。

关键符号：未识别

## 关键源码片段

### vllm/kernels/helion/ops/rms_norm_dynamic_per_token_quant.py

修复了 generate_inputs() 中 scale_ub 的计算，使用 RMS 归一化后加权输出的幅度均值与最大值的一半。

```
# In generate_inputs():
# Old: scale_ub = torch.mean(input).to(scale_dtype) # 零均值输入导致退化

# scale_ub clamps the per-token amax of the RMS-normed, weighted output.
# Use a non-degenerate upper bound (midway between the mean and max of
# that magnitude) so clamping is partially active and the baseline
# comparison is meaningful. torch.mean(input) ~= 0 for the zero-mean
# input would collapse every scale to the floor and saturate the output.
# Mirrors the reference normalization in baseline() below.
x = input.to(torch.float32)
residual.to(torch.float32)
rms = torch.rsqrt(x.pow(2).mean(dim=-1, keepdim=True) + epsilon)
x_norm_abs = ((x * rms).to(input.dtype) * weight).abs().to(torch.float32)
scale_ub = (0.5 * (x_norm_abs.mean() + x_norm_abs.amax())).to(scale_dtype)
```

### vllm/kernels/helion/ops/rms_norm_per_block_quant.py

修复了 generate_inputs() 中 scale_ub 的计算，与 rms_norm_dynamic_per_token_quant.py 类似但针对 per-block 量化。

```
# In generate_inputs():
# Old: scale_ub = torch.mean(input).to(scale_dtype)

# scale_ub clamps the per-group amax of the RMS-normed, weighted output.
# Use a non-degenerate upper bound (midway between the mean and max of
# that magnitude) so clamping is partially active and the baseline
# comparison is meaningful.
x = input.to(torch.float32)
residual.to(torch.float32)
rms = torch.rsqrt(x.pow(2).mean(dim=-1, keepdim=True) + epsilon)
x_norm_abs = ((x * rms).to(input.dtype) * weight).abs().to(torch.float32)
scale_ub = (0.5 * (x_norm_abs.mean() + x_norm_abs.amax())).to(scale_dtype)
```

### vllm/kernels/helion/ops/silu_and_mul_per_block_quant.py

修复了 generate_inputs() 中 scale_ub 的计算，使用 SiLU-and-mul 激活的幅度均值与最大值的一半。

```
# In generate_inputs():
# Old: scale_ub = torch.mean(input).to(scale_dtype)

# scale_ub clamps the per-group amax of the SiLU-and-mul activation. Use
# a non-degenerate upper bound (midway between the mean and max of the
# activation magnitude) so clamping is partially active and the baseline
# comparison is meaningful.
# Mirrors tests/kernels/helion/test_silu_and_mul_per_block_quant.py.
act_abs = SiluAndMul.forward_native(input.to(torch.float32)).abs()
scale_ub = (0.5 * (act_abs.mean() + act_abs.amax())).to(scale_dtype)
```

## 评论区精华

本 PR 无 review 评论，仅有两次 approve，变更较为直接。

- 暂无高价值评论线程

## 风险与影响

- 风险：风险极低：仅在测试输入生成器中修改了 scale_ub 的计算方式，不改变生产推理路径。所有修改均为测试辅助代码。
- 影响：影响范围仅限于 Helion kernel 的测试输入生成器，不涉及生产代码。修复后 autotuner 和正确性测试能够真正验证 scale_ub 路径的代码，避免假通过。
- 风险标记：测试基础设施变更

## 关联脉络

- 暂无明显关联 PR