PR #33854 完整报告

sgl-project/sglang

[diffusion] ERNIE-Image bit-exact fused RMSNorm+scale/shift (H200 1024^2 e2e 15.63 -> 15.00 s, denoise -3.3%)

合并时间：2026-08-06 19:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33854

执行摘要

- 一句话：ERNIE-Image 融合 RMSNorm+scale/shift 为 bit-exact Triton 内核，e2e 提速 4%
- 推荐动作：值得精读。核心亮点：① 如何通过复现浮点运算顺序（有序 fadd、禁用 FMA 收缩、inline PTX 复现 rsqrt.approx）实现 bit-exact 的 kernel fusion；② 运行时自校验 + 一次性 fallback 的稳健设计模式；③ 与 #33734 的递进关系，展示了如何把被否决的有损融合重新做成无损默认路径。对需要做 kernel fusion 且要求输出不变的场景有很强的借鉴意义。

功能与动机

PR #33734 已交付 bit-exact 的 residual-gate 融合，但明确放弃了两个 norm+scale/shift 融合（来自已关闭的 #30170），因为 CuTe-DSL 内核并非 bit-exact（同种子 PSNR 18.83 dB，远低于 25 dB 门槛）。本 PR 不是重新启用那些内核，而是用 Triton 重新实现并数值上逐步复刻 eager 链的取整过程，使其 bit-exact 并可无条件接入默认无损路径。每次 1024x1024 图像该块执行 7200 次 norm+modulate，eager 需 4-5 个 kernel，带宽受限，融合收益明显。

实现拆解

1. 新增 Triton 内核文件python/sglang/kernels/ops/diffusion/triton/rmsnorm_scale_shift_bitexact.py：实现 _rmsnorm_scale_shift_kernel（通过 HAS_GATE 分支同时支持普通 norm(x)*(1+scale)+shift 与 residual-gate 组合场景），并提供 _round_bf16_to_fp32、_mul_rn_f32、_rsqrt_approx_f32、_fold_adjacent 等数值辅助函数。内核逐位复刻 flashinfer CuTe RMSNormKernel 的归约顺序（per-fragment 有序 fp32 fadd 链、shfl.bfly 相邻对折叠树、rsqrt.approx）及 aten 链的 bf16 取整边界。内核以 custom op 注册，带 torch.compile-safe fake impl。
2. 模型侧接入python/sglang/multimodal_gen/runtime/models/dits/ernie_image.py：新增 _eager_norm_scale_shift、_ernie_norm_scale_shift、_ernie_gated_norm_scale_shift 包装函数。包装器在首次调用时通过 torch.equal 与 eager 链自校验，成功后置 _VERIFIED 标志；异常或比对失败则永久禁用快速路径并回退 eager；torch.compile 阶段不吞异常。在 ErnieImageSharedAdaLNBlock.forward 中将两处 adaLN 调用替换为包装器，第二处把 residual_gate_add_cuda + rmsnorm + modulate 合并为单核并同时返回

bit-identical 的 residual 流。

3. 测试配套：新增 test/registered/kernels/ops/diffusion/test_ernie_norm_scale_shift.py，覆盖真实 ERNIE 形状 (1,4216,4096)、CFG batch 形状 (2,1140,4096) 和 tpr=32 场景 (1,128,2048)，并断言快速路径确实被启用（_VERIFIED 为真且 _DISABLED 为假）。测试注册到 base-b-kernel-unit CI stage。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/ernie_image.py（模块 模型适配；类别 source；类型 core-logic；符号 _eager_norm_scale_shift, _ernie_norm_scale_shift, _ernie_gated_norm_scale_shift）：模型侧接入点，新增两个 bit-exact 融合 wrapper 并在 forward 中替换 eager 链，是融合进入默认路径的入口。
- python/sglang/kernels/ops/diffusion/triton/rmsnorm_scale_shift_bitexact.py（模块 内核实现；类别 source；类型 core-logic；符号 _rmsnorm_scale_shift_kernel, _round_bf16_to_fp32, _mul_rn_f32, _rsqrt_approx_f32）：新内核实现，是性能提升和 bit-exact 的核心，复刻 CuTe 归约顺序并复现 eager 取整边界。
- test/registered/kernels/ops/diffusion/test_ernie_norm_scale_shift.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_fused_norm_scale_shift_is_bit_exact）：验证 bit-exact 及快速路径确实启用，防止静默回退。

关键符号：_eager_norm_scale_shift, _ernie_norm_scale_shift, _ernie_gated_norm_scale_shift, _rmsnorm_scale_shift_kernel, _round_bf16_to_fp32, _mul_rn_f32, _rsqrt_approx_f32, _fold_adjacent, can_use_fused_rmsnorm_scale_shift, can_use_fused_scale_residual_rmsnorm_scale_shift

关键源码片段

python/sglang/multimodal_gen/runtime/models/dits/ernie_image.py

模型侧接入点，新增两个 bit-exact 融合 wrapper 并在 forward 中替换 eager 链，是融合进入默认路径的入口。

```
# 全局状态：区分“已禁用”（失败过）和“已验证”（首次自校验通过）
_ERNIE_FUSED_NORM_DISABLED = False
_ERNIE_FUSED_NORM_VERIFIED = False


def _ernie_norm_scale_shift(
    norm: RMSNorm, x: torch.Tensor, scale: torch.Tensor, shift: torch.Tensor
) -> torch.Tensor:
    """单核完成 ``norm(x) * (1 + scale) + shift``，与 eager 链逐位一致。

    Triton 内核复刻 flashinfer CuTe rmsnorm 的归约顺序和 aten 的每个
    bf16 取整边界。由于 bit-exact 取决于当前平台 ``RMSNorm.forward_cuda``
    分发到哪个 rmsnorm 实现，首次调用用 ``torch.equal`` 与 eager 链自校验，
    一旦不匹配就永久禁用快速路径。
    """
    global _ERNIE_FUSED_NORM_DISABLED, _ERNIE_FUSED_NORM_VERIFIED
```

```
    if (
        not _ERNIE_FUSED_NORM_DISABLED
        and norm.variance_size_override is None # 不支持 variance_override 场景
        and can_use_fused_rmsnorm_scale_shift(x, norm.weight, scale, shift)
        and (_ERNIE_FUSED_NORM_VERIFIED or not torch.compiler.is_compiling())
    ):
        try:
            out = fused_rmsnorm_scale_shift_bitexact(
                x, norm.weight, scale, shift, norm.variance_epsilon
            )
        except Exception as exc:
            # torch.compile 阶段不吞异常，避免 graph 捕获期出错被静默绕过
            if torch.compiler.is_compiling():
                raise
            logger.warning_once(f"Disabling ERNIE fused-norm fast path: {exc}")
            _ERNIE_FUSED_NORM_DISABLED = True
        else:
            if _ERNIE_FUSED_NORM_VERIFIED:
                return out
            # 首次调用：与 eager 链逐位比对
            ref = _eager_norm_scale_shift(norm, x, scale, shift)
            if torch.equal(out, ref):
                _ERNIE_FUSED_NORM_VERIFIED = True
                return out
            logger.warning_once(
                "ERNIE fused-norm fast path is not bit-exact against this "
                "platform's rmsnorm dispatch; falling back to eager"
            )
            _ERNIE_FUSED_NORM_DISABLED = True
            return ref

    return _eager_norm_scale_shift(norm, x, scale, shift)
```

python/sglang/kernels/ops/diffusion/triton/rmsnorm_scale_shift_bitexact.py

新内核实现，是性能提升和 bit-exact 的核心，复刻 CuTe 归约顺序并复现 eager 取整边界。

```
@triton.jit
def _rnms_norm_scale_shift_kernel(
    out_ptr, res_out_ptr,
    x_ptr, # norm 输入（无 gate）/ update（有 gate）
    residual_ptr, gate_ptr, weight_ptr, scale_ptr, shift_ptr,
    seq_len, eps,
    D: tl.constexpr, TPR: tl.constexpr, WPR: tl.constexpr,
    HAS_GATE: tl.constexpr,
):
    row = tl.program_id(0).to(tl.int64)
    batch = row // seq_len
    row_base = row * D
    vec_base = batch * D
```

```
# ----- pass 1: 以 CuTe 精确顺序求平方和 -----
# 复刻的"线程" tx 持有列 8*TPR*b + 8*tx + v；片段按 v 最快、b 其次
# 迭代，并组成一条有序 fadd 链（每个平方单独取整，禁止 FMA 收缩）。
tx = tl.arange(0, TPR) * 8
acc = tl.zeros((TPR,), dtype=tl.float32)
for b in tl.static_range(8):
    for v in tl.static_range(8):
        col = tx + (b * 8 * TPR + v)
        if HAS_GATE:
            rj = tl.load(residual_ptr + row_base + col).to(tl.float32)
            uj = tl.load(x_ptr + row_base + col).to(tl.float32)
            gj = tl.load(gate_ptr + vec_base + col).to(tl.float32)
            # eager 对：先对 gate*update 取整，再对加法结果取整
            xj = _round_bf16_to_fp32(rj + _round_bf16_to_fp32(gj * uj))
        else:
            xj = tl.load(x_ptr + row_base + col).to(tl.float32)
        acc = acc + _mul_rn_f32(xj, xj) # 不透明 mul.rn.f32，防止编译器收缩

# warp butterfly（偏移 1,2,4,8,16）等价于相邻对折叠树，
# 再将 WPR 个 warp 和用同样方式合并。
p = tl.reshape(acc, (WPR, 32))
p = _fold_adjacent(p, WPR, 16)
p = _fold_adjacent(p, WPR, 8)
p = _fold_adjacent(p, WPR, 4)
p = _fold_adjacent(p, WPR, 2)
p = _fold_adjacent(p, WPR, 1)
s = tl.reshape(p, (1, WPR))
if WPR == 2:
    s = _fold_adjacent(s, 1, 1)
rcp = tl.sum(_rsqrt_approx_f32(s / D + eps)) # 单元素，精确

# ----- pass 2: 归一化 + modulate，按 1024 列分块 -----
for i in tl.static_range(D // 1024):
    cols = i * 1024 + tl.arange(0, 1024)
    if HAS_GATE:
        r = tl.load(residual_ptr + row_base + cols).to(tl.float32)
        u = tl.load(x_ptr + row_base + cols).to(tl.float32)
        g = tl.load(gate_ptr + vec_base + cols).to(tl.float32)
        xin = _round_bf16_to_fp32(r + _round_bf16_to_fp32(g * u))
        tl.store(res_out_ptr + row_base + cols, xin)
    else:
        xin = tl.load(x_ptr + row_base + cols).to(tl.float32)
    w = tl.load(weight_ptr + cols).to(tl.float32)
    sc = tl.load(scale_ptr + vec_base + cols).to(tl.float32)
    sh = tl.load(shift_ptr + vec_base + cols).to(tl.float32)
    y = _round_bf16_to_fp32(xin * rcp * w) # (bf16)(x * rstd * w)
    one_plus = _round_bf16_to_fp32(1.0 + sc) # 复现 eager 的 1+scale 取整
    prod = _round_bf16_to_fp32(y * one_plus) # 复现 eager 的乘法取整
    tl.store(out_ptr + row_base + cols, prod + sh) # 存储时再取整到 bf16
```

评论区精华

本 PR 无 review 评论（仅一条 CI 链接评论）。PR body 中阐述了核心设计权衡：bit-exact 是“哪个 rmsnorm 实现在平台上分发”的属性，因此运行时必须用 torch.equal 自校验并在不匹配时永久回退 *eager*；同时明确拒绝使用有损的 CuTe-DSL 融合（同种子 PSNR 18.83 dB），而是重新实现数值上逐步复刻 *eager* 链的 Triton 内核。

- bit-exact 校验与 fallback 设计 (design): 采用首次调用自校验 + 一次性禁用快速路径，确保无损默认路径。

风险与影响

- 风险：数值假设耦合：内核复刻了 flashinfer CuTe RMSNormKernel 在 bf16 连续行、H == 64 * threads_per_row、cluster_n == 1 时的精确归约顺序，若未来 flashinfer 变更数值实现，自校验会检测到并回退 *eager*，但首次调用会多一次比对开销。 适用面受限：快速路径仅支持 bf16、连续 3D 张量、隐藏维度为 2048/4096/6144 等特定值，不满足条件时回退 *eager*，但覆盖了 ERNIE-Image 实际使用形状。 测试覆盖：当前仅 CUDA 单 GPU 测试，未覆盖 AMD/ 多 GPU 平台；但运行时自校验机制保证了跨平台安全性。 内核文件较大（342 行）且使用 inline PTX，维护门槛较高。
- 影响：影响范围限于 ERNIE-Image 模型（diffusion 管线）的默认生成路径，端到端耗时降低约 4%（H200 实测 15.63s -> 15.00s），*denoise* 阶段降约 3.3%。由于输出保持 bit-exact（整图 md5 与主分支一致），用户生成结果不受任何影响。对 sglang 其他模型与模块无影响。团队需关注后续 flashinfer 升级对该内核数值假设的影响，但自校验机制已提供安全保障。
- 风险标记：平台数值耦合，仅支持特定形状，运行时自校验保障

关联脉络

- PR #33734 [diffusion] ERNIE-Image bit-exact residual-gate fast path (H200 1024^2 *e2e* 16.17 -> 15.75 s): 本 PR 的延续，补完 #33734 放弃的两个 norm 融合。
- PR #30170 [diffusion] Fuse ERNIE AdaLN residual path: 被关闭的原始融合 PR，本 PR 以 bit-exact 方式重新实现其中两个被否决的融合。