PR #34318 完整报告

sgl-project/sglang

[Kernel] Route large SM90 row/column-scaled FP8 GEMMs to Torch

合并时间：2026-08-29 07:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34318

执行摘要

- 一句话：SM90 大 FP8 GEMM 路由到 Torch，显著提速但引发小形状回归
- 推荐动作：值得精读，尤其是 Fp8ScaledMMOp 的 BaseFusedOp 多后端注册与 _prefer_torch_rowwise_fp8 的资格检查模式，是未来内核选择机制的参考范本。但需注意启发式阈值必须经过广泛形状扫描验证，且应建立自动化的跨形状性能回归测试，防止此类硬编码规则再次引发性能回退。

功能与动机

PR body 明确指出：sgl_kernel.fp8_scaled_mm 在 SM90 的 MiniMax-H3 大 dense 形状上比 Torch _scaled_mm 慢，因此需要将更大形状路由到 Torch，同时保留 AOT 内核作为较小形状的回退，以获取 27%～34% 的 GEMM 延迟缩减并保持输出字节一致。

实现拆解

实现拆解如下：

1. 重构内核注册方式：在 python/sglang/kernels/ops/gemm/__init__.py 中，将原先直接 register_kernel 的 gemm.fp8_scaled_mm 改为继承 BaseFusedOp 的 Fp8ScaledMMOp 类，注册 AOT 与 TORCH 两个后端，其中 TORCH 能力门控为 SM90（CapabilityRequirement.cuda(min_sm=(9, 0), max_sm=(9, 0))），AOT 保持对所有 CUDA 架构通用。
2. 实现路由判定函数：新增 _prefer_torch_rowwise_fp8，先做硬性资格检查（设备、dtype、布局、无 bias、row/column 缩放形状），再按调参后的形状阈值 (k >= 5376 and n >= 3584) or (k >= 3584 and m >= 8192) 决定是否走 Torch。该阈值基于 H100 上 MiniMax-H3 全部 64 个 dense 形状的实测扫描，保证每个形状都选中测量更快的实现。
3. 运行时接线：在 python/sglang/srt/layers/quantization/fp8_utils.py 中将 fp8_scaled_mm 的导入从 sgl_kernel 改为 sglang.kernels.ops.gemm，使 SRT 量化层统一走新路由，而不是直接调 AOT 内核。
4. 测试配套：更新 test/registered/kernels/ops/layernorm/test_kernels_namespace.py 中注册后端集合（fp8_scaled_mm 从 {"aot"} 变为 {"aot", "torch", "torch_compile"}），新增 test_fp8_scaled_mm_requires_explicit_registry_backend 验证 SM90 多后端时必须显式指定 backend，并将原单一后端选择测试改用 kvcache.reshape_and_cache_flash。

5. 基准验证：PR 中给出 H100 上 64 形状扫描、MiniMax-H3 8×H100 端到端对比，以及 multimodal_gen 全套手动验证结果。

关键文件：

- python/sglang/kernels/ops/gemm/__init__.py（模块 内核路由；类别 infra；类型 infrastructure；符号 _prefer_torch_rowwise_fp8, Fp8ScaledMMOp, backend_eligible, forward_native）：核心变更文件：将 fp8_scaled_mm 从单一 AOT 注册重构为 BaseFusedOp 多后端路由，新增 Torch 后端与形状启发式判定。
- test/registered/kernels/ops/layernorm/test_kernels_namespace.py（模块 内核测试；类别 test；类型 test-coverage；符号 test_fp8_scaled_mm_requires_explicit_registry_backend）：更新内核注册表断言，验证 fp8_scaled_mm 变为多后端后必须显式指定 backend，防止调用方隐式选择出错。
- python/sglang/srt/layers/quantization/fp8_utils.py（模块 量化层；类别 source；类型 dependency-wiring）：运行时接线：将 fp8_scaled_mm 导入从 sgl_kernel 改为 sglang.kernels.ops.gemm，使 SRT 量化层统一走新路由。

关键符号：_prefer_torch_rowwise_fp8, Fp8ScaledMMOp.backend_eligible, Fp8ScaledMMOp.forward_native, Fp8ScaledMMOp.forward_aot, test_fp8_scaled_mm_requires_explicit_registry_backend

关键源码片段

python/sglang/kernels/ops/gemm/__init__.py

核心变更文件：将 fp8_scaled_mm 从单一 AOT 注册重构为 BaseFusedOp 多后端路由，新增 Torch 后端与形状启发式判定。

```
# 判断当前 FP8 GEMM 形状是否应优先走 Torch 的 SM90 NVJet 内核。
def _prefer_torch_rowwise_fp8(
    mat_a: torch.Tensor,
    mat_b: torch.Tensor,
    scales_a: torch.Tensor,
    scales_b: torch.Tensor,
    out_dtype: torch.dtype,
    bias: Optional[torch.Tensor],
) -> bool:
    # 先做硬性资格过滤：仅 CUDA、同设备、支持 torch._scaled_mm ...
    if (
        mat_a.device.type != "cuda"
        or mat_b.device != mat_a.device
        or not hasattr(torch, "_scaled_mm")
        or out_dtype != torch.bfloat16
        or bias is not None
        or mat_a.dtype != torch.float8_e4m3fn
        or mat_b.dtype != torch.float8_e4m3fn
        or mat_a.ndim != 2
        or mat_b.ndim != 2
        or mat_a.stride(1) != 1
```

```
        or mat_b.stride(0) != 1
    ):
        return False

    m, k = mat_a.shape
    n = mat_b.shape[1]
    # 本路径只支持 row/column 级缩放：A 每行一个独立 FP32 scale，B 每列一个。
    if (
        scales_a.dtype != torch.float32
        or scales_b.dtype != torch.float32
        or scales_a.device != mat_a.device
        or scales_b.device != mat_a.device
        or not scales_a.is_contiguous()
        or not scales_b.is_contiguous()
        or scales_a.numel() != m
        or scales_b.numel() != n
    ):
        return False

    # 形状启发式在 H100 上针对 MiniMax-H3 的全部 64 个 dense 形状调参：
    # 该选择器对每个形状都命中了实测更快的实现，同时用较小的 K 保留 AOT。
    return (k >= 5376 and n >= 3584) or (k >= 3584 and m >= 8192)

# FP8 GEMM 的融合算子注册：优先 AOT（sgl_kernel），满足条件时落到 Torch。
class Fp8ScaledMMOp(BaseFusedOp):
    """FP8 GEMM：A 按行、B 按列各自独立缩放。"""

    op = "gemm.fp8_scaled_mm"
    priority = (KernelBackend.AOT, KernelBackend.TORCH)
    capabilities = {
        KernelBackend.AOT: _CUDA, # AOT 对所有 CUDA 架构通用
        KernelBackend.TORCH: _SM90, # Torch 路径仅限 SM90（调参硬件）
    }
    # backend_eligible / forward_aot / forward_native 分别实现
    # 资格检查、AOT 回调和 Torch 前向，择路由 _prefer_torch_rowwise_fp8 决定。
```

评论区精华

review 过程中的核心讨论集中在两点：

1. 测试触发覆盖问题：mickqian 指出 multimodal_gen 测试未自动触发，RunFMe 随后在 H200 节点手动运行了完整的多模态生成测试套件并贴出详细结果，确认无回归。
2. 合并后暴露的路由启发式回归：hnyls2002 在合并后报告 test_w8a8_quantization.py::TestW8A8Fp8.test_throughput（Llama-3.1-8B-Instruct-FP8-dynamic，bs=1）从约 215 tok/s 降到约 190 tok/s，跌破 200 tok/s 阈值，原因是路由条件 (k >= 5376 and n >= 3584) 没有考虑 M 太小的情况，Torch 在小 M decode 形状上反而更慢。RunFMe 承认问题并立即在 PR #37018 中增加 M 门控修复。

- multimodal_gen 测试未自动触发 (testing): 作者手动验证通过，但 CI 覆盖仍有缺口。

- 路由启发式引发 W8A8 FP8 *decode* 性能回退 (performance): RunFMe 确认问题并在 PR #37018 中增加 M 门控修复。
- 评审确认与多架构影响范围 (question): 评审通过，关注点在于路由对非 H3 模型的潜在影响。

风险与影响

- 风险：主要风险集中在路由启发式的通用性：
- 小形状性能回退：硬编码阈值针对 MiniMax-H3 调参，未覆盖小 M 的 *decode* 场景，已实际引发 W8A8 FP8 *decode* 性能回退（215→190 tok/s），虽已由 #37018 修复，但说明启发式规则脆弱，未来新模型形状可能再次误伤。
- 仅 SM90 验证：PR 明确声明只在 SM90 硬件调参，SM100/SM120 上 Torch _scaled_mm 的胜负未知，后续架构可能因沿用相同规则而选错后端。
- 后端能力门控依赖注册表：gemm.fp8_scaled_mm 从单后端变为多后端，任何依赖 select_kernel 隐式选择的调用点在 SM90 上都会因多后端而抛出 ValueError，需要调用方显式指定 backend，存在遗漏调用点的风险。
- 测试覆盖缺口：multimodal_gen 测试未自动触发，依赖作者手动验证，CI 防护不足。
- 影响：影响范围包括：所有在 SM90 上使用 FP8 行 / 列缩放 GEMM 的模型（尤其 MiniMax-H3）可获得 14%～34% 的 GEMM 延迟缩减；SRT 量化层 fp8_utils.py 统一走新路由，小幅影响所有 FP8 量化模型；内核注册表行为变化使 fp8_scaled_mm 无法隐式选择，需要调用方适配；对团队而言，引入了一个可复用的多后端路由模式，但也暴露了启发式硬编码的维护成本。
- 风险标记：启发式路由误伤小形状，已引发 W8A8 性能回归，仅 SM90 调参验证，多后端选择需显式指定，multimodal_gen 测试未自动触发

关联脉络

- PR #33275 Add MiniMax-H3 and its online FP8 transformer path: 本 PR 的 follow-up：在 MiniMax-H3 引入后，进一步为大形状 FP8 GEMM 增加 Torch 路由以提升性能。
- PR #37018 Fix FP8 rowwise routing cutoffs to exclude smaller shapes: 修复本 PR 引发的 W8A8 *decode* 性能回退，通过增加 M 门控收窄路由条件。