PR #35505 完整报告

sgl-project/sglang

[Deepseek-V4] Enable shared-experts fusion on the flashinfer_mxfp4 (trtllm-gen) MoE path

合并时间：2026-08-26 05:55

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35505

执行摘要

- 一句话：启用 DSV4 flashinfer_mxfp4 共享专家融合，显著降低 TTFT 与 ITL
- 推荐动作：值得精读。这是一个 " 机制已存在、本 PR 负责启用与防护 " 的典型范例：源码改动极小（12 行 guard），但通过契约测试把量化 wire format 钉死，并通过 fail-fast 防护避免 EP 下静默错误。可重点关注 uses_per_rank_fused_shared_slots() 与 _maybe_load_fp8_shared_expert_as_fp4 的交互，以及测试中对 -0.0 编码的断言处理方式。

功能与动机

DeepSeek-V4-Flash 在 flashinfer_mxfp4（trtllm-gen）路径上，共享专家默认作为独立 FP8 MLP 在另一条 stream 上运行。PR body 明确说明：融合后 'the shared expert is routed as expert slot 256 through the same MoE kernel, so the whole MoE runs on a single stream instead of a separate FP8 MLP on an alternate stream (~4 fewer kernel launches + 2 fewer stream syncs per layer)'。底层机制（FP8→MXFP4 加载期 requant、slot 路由）由 #27349 提供，本 PR 负责启用、验证，并补上关键正确性防护：EP 下没有 per-rank shared slots 时融合会静默计算错误，必须提前拒绝。

实现拆解

1. EP 防护（python/sglang/srt/models/deepseek_v4.py）：在 shared_experts_fusion_disable_reason classmethod 中新增分支，当 get_parallel().moe_ep_size > 1 且 uses_per_rank_fused_shared_slots() 为 False 时，返回明确的 disable reason，使 loader 在构建任何层之前 fail-fast。原因：EP 下每个 rank 只持有 routed experts 的一个切片，fusion slot 无法追加到 routed weight 张量上，静默融合会算出错误结果。同时补充 uses_per_rank_fused_shared_slots 的 import。
2. 复用既有 requant 管线（无额外源码改动）：load_weights 将 mlp.shared_experts.* 重命名为 mlp.experts.{n_routed}.*，FusedMoE._maybe_load_fp8_shared_expert_as_fp4（#27349 引入）按 128x128 block 将原始 FP8 权重与其 block scale 通过 quantize_block_fp8_weight_to_mxfp4 转为 MXFP4 并写入融合槽位。本 PR 不修改 trtllm-gen kernel，仅启用已验证的能力。
3. 契约测试（新增 test/registered/unit/models/test_deepseek_v4_mxfp4_shared_expert_requant.py）：使用 torch 参考实现 _dequant_mxfp4 反量化，验证 nibble 顺序（低 nibble = 偶数列）、符号位（code bit 3）、e8m0 偏置 127、block scale 应用与往返误差

上限（< 0.15），注册到 base-a-test-cpu suite。测试还揭示量化器把 0.0 编码为 -0.0（code 8），断言改为只验证 padding 反量化后为 0，不锁定符号位。

4. 文档配套（docs/cookbook/autoregressive/DeepSeek/DeepSeek-V4.mdx）：在 Configuration Tips 下新增 "Shared experts fusion (Blackwell, flashinfer_mxfp4)" 小节，说明 --enforce-shared-experts-fusion 用法、收益数据（QPS 1-8 下 TTFT -13% 至 -21%、P99 ITL -15% 至 -53%）与不适用 EP 的约束；recipe 命令行按 reviewer 要求保持不变。

关键文件：

- python/sglang/srt/models/deepseek_v4.py（模块 模型层；类别 source；类型 core-logic；符号 shared_experts_fusion_disable_reason, uses_per_rank_fused_shared_slots）：核心源码变更：在 shared_experts_fusion_disable_reason 中新增 EP 下无 per-rank shared slots 时的 fail-fast 防护，避免融合在 EP 场景静默算错。
- test/registered/unit/models/test_deepseek_v4_mxfp4_shared_expert_requant.py（模块 单元测试；类别 test；类型 test-coverage；符号 _dequant_mxfp4, decode, TestQuantizeBlockFp8WeightToMxfp4, _requant）：新增 CPU 单元测试，pin 住加载期 FP8→MXFP4 共享专家 requant 的字节级 packing/scale 契约，是防止未来 kernel 或量化器 layout 漂移的关键保障。
- docs/cookbook/autoregressive/DeepSeek/DeepSeek-V4.mdx（模块 使用文档；类别 other；类型 documentation）：补充 Shared experts fusion 使用文档，说明启用方式、性能收益与 no-EP 约束；recipe 命令行按 reviewer 要求保持原样。

关键符号：shared_experts_fusion_disable_reason, uses_per_rank_fused_shared_slots, quantize_block_fp8_weight_to_mxfp4, _maybe_load_fp8_shared_expert_as_fp4, _dequant_mxfp4, TestQuantizeBlockFp8WeightToMxfp4

关键源码片段

python/sglang/srt/models/deepseek_v4.py

核心源码变更：在 shared_experts_fusion_disable_reason 中新增 EP 下无 per-rank shared slots 时的 fail-fast 防护，避免融合在 EP 场景静默算错。

```
@classmethod
def shared_experts_fusion_disable_reason(cls, hf_config, quant_config):
    """V4 只在显式要求时融合，且 checkpoint 必须恰好携带一个共享专家。
    该方法由 loader 在任何层构建之前调用。"""
    # 量化精度不匹配时，即使指定 --enforce-shared-experts-fusion 也要禁用
    # （共享专家精度高于 routed experts，无法并入量化路由路径）
    if quant_blocks_shared_experts_fusion(quant_config):
        return (
            "Quantization keeps shared experts at a higher precision than the "
            "routed experts, so they cannot be fused into the quantized "
            "routed-expert path."
        )
    # 本 PR 新增：EP 下每个 rank 只持有 routed experts 的一个切片，
    # 融合槽位无法追加到 routed weight 张量，强行融合会静默算错；
```

```
    # 只有 DeepEP / MegaMOE 的 per-rank shared slots 支持 EP 下融合
    if get_parallel().moe_ep_size > 1 and not uses_per_rank_fused_shared_slots():
        return (
            "Expert parallelism keeps only a slice of the routed experts on "
            "each rank, so the fused shared expert cannot be appended to the "
            "routed weight tensor (only DeepEP/MegaMOE per-rank shared slots "
            "support fusion under EP)."
        )
    if not get_exec().moe.enforce_shared_experts_fusion:
        return "Config does not support fused shared expert(s)."
    if hf_config.n_shared_experts != 1:
        raise ValueError(
            "DeepSeek V4 shared-experts fusion expects exactly one shared "
            f"expert, but got n_shared_experts={hf_config.n_shared_experts}."
        )
    return None
```

test/registered/unit/models/test_deepseek_v4_mxfp4_shared_expert_requant.py

新增 CPU 单元测试，pin 住加载期 FP8→MXFP4 共享专家 requant 的字节级 packing/scale 契约，是防止未来 kernel 或量化器 layout 漂移的关键保障。

```
  def _dequant_mxfp4(packed: torch.Tensor, scales: torch.Tensor) -> torch.Tensor:
      """参考解量化 MXFP4 布局，与 trtllm-gen kernel 及 DSV4 checkpoint
      的 routed experts 消费的布局一致：低 nibble 为偶数列 code，
      符号位在 code bit 3，每 32 个同行元素共享一个 e8m0 scale。"""
      as_u8 = packed.view(torch.uint8)
      codes_lo = (as_u8 & 0x0F).long() # 低位 nibble 对应偶数列
      codes_hi = (as_u8 >> 4).long() # 高位 nibble 对应奇数列

      def decode(codes):
          magnitudes = _E2M1_LUT[codes & 0x7] # bits 2:0 索引幅度表
          return torch.where(codes >= 8, -magnitudes, magnitudes) # bit 3 是符号位

      vals = torch.stack([decode(codes_lo), decode(codes_hi)], dim=-1)
      vals = vals.reshape(packed.shape[0], packed.shape[1] * 2)
      exponents = scales.view(torch.uint8).float() - 127.0 # e8m0 偏置 127
      return vals * torch.pow(2.0, exponents).repeat_interleave(32, dim=-1)


  def test_packing_layout_contract(self):
      # 锁定 quantize_block_fp8_weight_to_mxfp4 产出的字节布局：低 nibble 是
      # 偶数列，高 nibble 是奇数列，e8m0 scale 偏置为 127
      w = torch.zeros(1, 32, dtype=torch.bfloat16)
      w[0, 0] = 0.5 # 期望 code 1
      w[0, 1] = -3.0 # 幅度索引 5 + 符号位 -> code 13
      w[0, 2] = 6.0 # 期望 code 7
      w[0, 3] = 1.5 # 期望 code 3
      packed, scales = self._requant(w)
```

```
        self.assertEqual(packed.dtype, torch.int8)
        self.assertEqual(packed.shape, (1, 16)) # 32 列 4bit 打包为 16 字节
        self.assertEqual(scales.dtype, torch.float8_e8m0fnu)
        self.assertEqual(scales.shape, (1, 1))
        # group amax 6.0 -> scale 2**0 -> 偏置后的 e8m0 指数为 127
        self.assertEqual(scales.view(torch.uint8)[0, 0].item(), 127)
        as_u8 = packed.view(torch.uint8)
        # 第一字节 = code(0.5) | code(-3.0) << 4；低 nibble 是偶数列
        self.assertEqual(as_u8[0, 0].item(), 0x01 | (0x0D << 4))
```

评论区精华

核心讨论围绕两点：一是 Fridge003 反对修改 cookbook 的 recipe 命令，要求改为在 configuration tips 中描述 shared expert fusion 用法，shikicloud 据此新增了专门小节；二是 Fridge003 追问共享专家的加载与 requant 处理位置，shikicloud 说明该路径是 #27349 已经存在的 FusedMoE._maybe_load_fp8_shared_expert_as_fp4，新单元测试 pin 住其字节级输出格式。此外，CPU 单元测试首次 CI 运行失败，暴露量化器将 0.0 编码为 -0.0（code 8）的细节，作者修改断言后 rerun 通过。

- cookbook recipe 命令不应被修改，应改为文档说明 (documentation): shikicloud 在 DeepSeek-V4.mdx 的 configuration-tips 下新增 "Shared experts fusion (Blackwell, flashinfer_mxfp4)" 小节，recipe 命令保持原样。
- 共享专家的加载与 requant 在哪里处理 (question): 确认无需修改 kernel 或加载路径，本 PR 只启用并新增契约测试。
- requant 测试对 0.0 编码的断言修复 (testing): 提交 fc03ea09 修复断言，/rerun-test 通过。

风险与影响

- 风险：
  1. EP 场景正确性：新增 guard 依赖 uses_per_rank_fused_shared_slots() 对 DeepEP/MegaMOE per-rank shared-slot 路径的准确识别；若该判断有边界遗漏，显式启用 fusion 的 EP 用户会被 fail-fast 拒绝（而非静默算错），属于安全侧失败。
  2. 数值精度：加载期 FP8→MXFP4 requant 是有损压缩，单元测试限定往返相对误差 < 0.15，GB200 tp4 上 gsm8k 与 AIME25 精度与基线持平，但其他模型规模或 batch 形态下误差上限仍需观测。
  3. 兼容性：本特性仅对 Blackwell + flashinfer_mxfp4 后端生效，其他 MoE 后端行为不变；文档明确标注 no-EP 约束，避免误用。
  4. 测试契约敏感性：新增测试把 nibble 顺序、符号位、e8m0 偏置钉死为契约，未来若 trtllm-gen 或量化器调整 layout 会导致测试失败，这是有意的保护，但也意味着测试需要与该 kernel 保持同步演进。 - 影响：用户侧：GB200 + flashinfer_mxfp4 部署 DeepSeek-V4-Flash 的用户可直接通过 --enforce-shared-experts-fusion 获得 13%～21% 的 TTFT 改善和 15%～53% 的 P99 ITL 改善，吞吐持平；EP 用户会被明确拒绝启用，避免之前可能存在的静默错误。系统侧：整个 MoE 收敛到单 stream，减少 stream sync 带来的调度开销，同时减少 kernel launch 数量，对低延迟场景（QPS 1-8）收益明显。团队侧：新增的字节级契约测试为后续修改 FP8/MXFP4 加载路径提供了安全

网，文档补充降低了误用概率。 - 风险标记：EP 下静默错误风险（已加 fail-fast 防护），FP8→MXFP4 requant 有损，仅 Blackwell flashinfer_mxfp4 生效，契约测试曾因-0.0 编码失败

关联脉络

- PR #36097 Fix MXFP8 MoE weight sizing for non-gated models: 同为 MXFP8/MoE 量化契约类修复，说明该区域 layout 与尺寸契约容易出错，本 PR 的字节级契约测试与此呼应。
- PR #35630 [AMD] Enable Mori-EP on kimi-k3: 涉及 EP 与 MXFP4 量化路径的组合场景，与本 PR 新增的 EP guard 处于同一问题域。
- PR #35188 [Bugfix] Fix int32 destination offset overflow in CUTLASS MoE pre-reorder: 同为 MoE 内核路径稳定性的修复，共享专家融合后 slot 256 的路由行为与 MoE kernel 的索引 / 偏移处理相关。