PR #35739 完整报告

sgl-project/sglang

[multimodal] Fix NVFP4 diffusion models on sm_120 (RTX PRO 6000 / RTX 50xx)

合并时间：2026-08-29 14:42

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35739

执行摘要

- 一句话：修复 NVFP4 扩散模型在 sm_120 上不可用的问题
- 推荐动作：值得精读，特别是理解默认后端选择逻辑和 TMA 布局的必要性。该 PR 体现了对硬件能力差异的精细化处理，以及删除不可达代码的整洁重构，适合作为后续支持新 GPU 架构的参考。

功能与动机

NVFP4 扩散检查点在 sm_120 GPU 上完全无法服务。第一个缺陷是默认的 FP4 GEMM 后端 'trtllm' 在 sm_120 上不受支持，导致 flashinfer.mm_fp4 抛错。第二个缺陷是 block scale 布局选择逻辑存在不可达分支，导致部分模型（非 FLUX.1 前缀）静默生成错误输出。

实现拆解

1. 修改 CudaPlatform.get_modelopt_flashinfer_fp4_backend()：将默认后端从固定 "trtllm" 改为根据 is_sm120() 动态选择：sm_120 返回 "auto"，否则 "trtllm"，并保留环境变量覆盖机制。
2. 在 ModelOptFp4LinearMethod.process_weights_after_loading() 中，删除基于模块名前缀的启发式分支，无条件应用 128x4 TMA reshape 和 permute，因为该函数剩余路径只用于 TMA 布局消费的后端。
3. 新增单元测试文件 test_modelopt_fp4_backend.py，通过 mock 环境变量和 is_sm120 状态覆盖各种场景，验证默认值和显式配置的解析。

关键文件：

- python/sglang/multimodal_gen/runtime/platforms/cuda.py（模块 平台适配；类别 source；类型 core-logic；符号 get_modelopt_flashinfer_fp4_backend）：核心逻辑修改，根据 GPU 能力选择默认 FP4 后端，是问题 1 的修复点。
- python/sglang/multimodal_gen/runtime/layers/quantization/modelopt_quant.py（模块 量化层；类别 source；类型 data-contract；符号 process_weights_after_loading）：修复 block scale 布局选择缺陷，删除不可达分支，确保 TMA 布局被正确应用。
- python/sglang/multimodal_gen/test/unit/test_modelopt_fp4_backend.py（模块 测试；类别 test；类型 test-coverage；符号 _backend, TestModeloptFp4BackendDefault）：新增的单元测试，确保后端默认逻辑在各种 GPU 和显式配置下正确。

关键符号：CudaPlatform.get_modelopt_flashinfer_fp4_backend，ModelOptFp4LinearMethod.process_weights_after_loading，_backend

关键源码片段

python/sglang/multimodal_gen/runtime/platforms/cuda.py

核心逻辑修改，根据 GPU 能力选择默认 FP4 后端，是问题 1 的修复点。

```
@classmethod
@lru_cache(maxsize=1)
def get_modelopt_flashinfer_fp4_backend(cls) -> str:
    backend = envs.SGLANG_DIFFUSION_FLASHINFER_FP4_GEMM_BACKEND
    # flashinfer.mm_fp4 rejects backend="trtllm" on sm_120 ("does not support
    # backend 'trtllm' with capability 120"); "auto" resolves to its sm_12x
    # NVFP4 kernel there.
    default_backend = "auto" if cls.is_sm120() else "trtllm"
    if backend is None:
        return default_backend

    # 显式设置时，统一小写并映射 flashinfer_* 前缀，非法值回退到默认后端。
    backend = backend.lower()
    backend = {
        "flashinfer_cudnn": "cudnn",
        "flashinfer_cutlass": "cutlass",
        "flashinfer_trtllm": "trtllm",
        "trtllm": "trtllm",
        "cudnn": "cudnn",
        "auto": "auto",
    }.get(backend, backend)
    if backend not in {"auto", "cudnn", "cutlass", "trtllm"}:
        logger.warning(
            "Unsupported SGLANG_DIFFUSION_FLASHINFER_FP4_GEMM_BACKEND=%r. "
            "Falling back to %r.",
            backend,
            default_backend,
        )
        return default_backend
    return backend
```

python/sglang/multimodal_gen/runtime/layers/quantization/modelopt_quant.py

修复 block scale 布局选择缺陷，删除不可达分支，确保 TMA 布局被正确应用。

```
def process_weights_after_loading(self, layer: torch.nn.Module) -> None:
    ...
    scale_ndim = scales.ndim
    if scale_ndim == 2:
        scales = scales.unsqueeze(0)
    assert scales.ndim == 3
```

```
    B, M, K = scales.shape
    M_padded = round_up(M, 128)
    K_padded = round_up(K, 4)
    padded_scales = torch.zeros((B, M_padded, K_padded), dtype=scales.dtype)
    padded_scales[:B, :M, :K] = scales

    # 所有在此处可达的 FP4 GEMM 都按 128x4 TMA 布局读取 block scale；
    # 唯一需要自己 shuffle 布局的 trtllm 后端已在上面提前 return。
    padded_scales = padded_scales.reshape(
        B, M_padded // 128, 4, 32, K_padded // 4, 4
    )
    padded_scales = padded_scales.permute(0, 1, 4, 3, 2, 5)

    padded_scales = padded_scales.contiguous().cuda()
    padded_scales = (
        padded_scales.reshape(M_padded, K_padded)
        if scale_ndim == 2
        else padded_scales.reshape(B, M_padded, K_padded)
    )
    copy_or_rebind_param(layer, "weight_scale_interleaved", padded_scales)
```

评论区精华

主要讨论集中在 CI 失败项上：作者 whn09 解释红色 CI 通道（amd-rocm720、npu-a3、pr-test-extra）是因为该 PR 只修改了 platforms/cuda.py 和 ModelOpt FP4 加载路径，这些路径在 ROCm 或 NPU 上不会执行，因此失败与本次改动无关。BBuf 提供了额外验证：在 RTX PRO 6000 Blackwell 上使用官方 NVFP4 checkpoint 进行了真实推理验证，确认输出与之前环境变量 workaround 结果 bit-identical。

- CI 失败是否与本次改动有关 (other): CI 失败被认定为无关，未阻碍合并。
- 基于真实 checkpoints 的验证 (testing): 验证通过，增强了修复可信度。

风险与影响

- 风险：
  1. 修改了默认后端逻辑，可能影响 sm_120 以外的 GPU（但默认仍为 trtllm，无变化）。
  2. 删除了启发式分支，若存在依赖 FLUX.1 前缀来判断布局的隐藏路径，可能导致回归，但注释已说明该分支实为不可达。
  3. 新增测试依赖 mock is_sm120，若未来 is_sm120 实现变化，测试可能需要调整。整体风险较低。 - 影响：影响范围主要限于 NVFP4 量化扩散模型在 sm_120 GPU 上的服务能力，属于明确的功能修复。对于其他平台和模型，保持原有行为，无性能影响。团队收益是消除了用户不得不手动设置环境变量的 workaround，提升易用性。 - 风险标记：默认行为变更，存在不可达逻辑删除，缺少文档更新

关联脉络

- PR #35740 MiniMax-H3 model-side quantized-scale fixes: PR body 指出该 PR 与 #35740 共同协作才能让 NVFP4 MiniMax-H3 工作，二者层级独立。