PR #35353 完整报告

sgl-project/sglang

[diffusion] make --vae-tiling honest, fix the decode OOM advice, gate NVFP4 on Blackwell

合并时间：2026-08-19 09:27

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35353

执行摘要

- 一句话：VAE tiling/OOM 提示修正，NVFP4 增加卡型门禁
- 推荐动作：值得精读。本 PR 展示了三条可复用的工程方法：一是把 " 静默吞异常 " 改造为 " 诚实告警 " 且不破坏现有调用；二是修建议类日志前先做因果归因（区分权重移动与激活值溢出）；三是用 " 能力不可知时放行 " 的保守策略实现硬件门禁，既解决虚假表象又避免误伤。重点关注 decoding.py 的异常守卫缩窄论证和 ModelOptFp4LinearMethod.__init__ 的门禁设计。

功能与动机

作者在 PR body 中明确写道："Three failures that all present as something other than what they are. Found while auditing consumer-GPU support." 具体动机包括：--vae-tiling 对无 enable_tiling 的 VAE 是静默 no-op，MiniMax-H3 的 VAE 恰好因 decoder_tiling 由配置置位而照常 tiling，导致用户以为参数生效；decode OOM 提示推荐 --vae-cpu-offload，但该参数只移动权重、对溢出的激活值无效，真正杠杆是释放已空闲的 DiT 与 encoder;ModelOptFp4Config.get_min_capability() 声明最低算力 100 却从未被调用，pre-Blackwell 卡在加载成功后才会在 CUDA kernel 内以不透明错误失败。

实现拆解

1. 解码阶段入口（python/sglang/multimodal_gen/runtime/pipelines_core/stages/decoding.py 的 decode 方法）：将原来 try: self.vae.enable_tiling() except Exception: pass 的守卫缩窄为仅捕获 AttributeError，并对不支持的 VAE 打印警告、点名 VAE 类名，告知其 tiling 行为由 VAE 配置决定；同时重写两条 OOM logger.warning，先建议 --cpu-offload-components dit,text_encoder 释放已结束的组件，再引导降低 VAE 配置中的 tile size，最后才降低分辨率或帧数。
2. MiniMax-H3 VAE 适配（python/sglang/multimodal_gen/runtime/models/vaes/minimax_h3_video_vae/klvae.py）：新增 enable_tiling() 方法，仅设置 self.decoder_tiling = True。因为 _adaptive_decode 每次解码都会读取该标志，所以新的 tiling 开关对下一次 decode 立即生效；已从 VAE 配置置位 tiling 的模型不受影响，运行时开关首次能真正触达这类 VAE。
3. NVFP4 能力门禁（python/sglang/multimodal_gen/runtime/layers/quantization/modelopt_quant.py）：在 ModelOptFp4LinearMethod.__init__ 中调用 current_platform.get_device_capability() 并与 quant_config.get_min_capability() 比较

，低于 10.0 即抛出带指引的 RuntimeError；当 capability 读取为 None 时保持原行为放行，确保门禁只把 " 崩溃 " 变成 " 消息 "，不会阻塞原本可用的卡。

4. 测试配套：新增 test_modelopt_fp4_capability_gate.py，通过 unittest.mock.patch 覆盖 8.6/8.9/9.0 拒绝、10.0/12.0 放行、None 放行三条分支；修改 test_transformer_quant.py 与 test_ideogram4.py，在构造 NVFP4 量化模型的用例中注入 DeviceCapability(10, 0) 作为 mock，使测试在非 Blackwell CI 机器上也能通过新门禁。

关键文件：

- python/sglang/multimodal_gen/runtime/layers/quantization/modelopt_quant.py（模块 量化层；类别 source；类型 data-contract；符号 ModelOptFp4LinearMethod）：NVFP4 门禁核心实现：在 ModelOptFp4LinearMethod 构造期强制执行 get_min_capability() 声明的 Blackwell 最低算力，把 kernel 内崩溃提前为可读 RuntimeError，是三个修复中影响面最广的量化加载路径变更。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/decoding.py（模块 解码阶段；类别 source；类型 core-logic；符号 DecodingStage.decode）：VAE 解码主路径：收窄 enable_tiling 异常守卫为 AttributeError 并显式告警，同时重写两条 OOM 建议，让 --vae-tiling 与 OOM 提示从 " 表象 " 回归 " 本质 "。
- python/sglang/multimodal_gen/runtime/models/vaes/minimax_h3_video_vae/klvae.py（模块 VAE 模块；类别 source；类型 data-contract；符号 enable_tiling）：MiniMax-H3 VAE 是 --vae-tiling 静默失效的典型受害对象，新增 enable_tiling 方法使运行时开关能真正触达这类 VAE，并让新告警逻辑有明确的适配入口。
- python/sglang/multimodal_gen/test/unit/test_modelopt_fp4_capability_gate.py（模块 量化测试；类别 test；类型 test-coverage；符号 _config, TestModelOptFp4CapabilityGate, test_rejects_pre_blackwell_with_an_actionable_message, test_allows_blackwell）：新增单元测试，覆盖 NVFP4 门禁的拒绝、放行、能力不可知三分支，是本次量化 gate 的回归保护核心。
- python/sglang/multimodal_gen/test/unit/test_transformer_quant.py（模块 量化测试；类别 test；类型 test-coverage；符号 test_flux_single_transformer_block_modelopt_excludes_use_full_prefix）：FLUX 单 transformer block 的 NVFP4 用例需要 mock Blackwell capability，否则在非 Blackwell CI 机器上会被新门禁拦截。
- python/sglang/multimodal_gen/test/unit/test_ideogram4.py（模块 图像模型；类别 test；类型 test-coverage；符号 test_ideogram_dit_nvfp4_quant_config_uses_native_fp4_linears, test_ideogram_dit_tp_nvfp4_uses_megatron_parallel_quant_linears）：Ideogram4 的 NVFP4 DiT 与 TP 量化用例同样需要 mock Blackwell capability，保持 CI 在各类 GPU 上可稳定通过。

关键符号：ModelOptFp4LinearMethod.init, DecodingStage.decode, enable_tiling, _adaptive_decode

关键源码片段

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

NVFP4 门禁核心实现：在 ModelOptFp4LinearMethod 构造期强制执行 get_min_capability() 声明的 Blackwell 最低算力，把 kernel 内崩溃提前为可读 RuntimeError，是三个修复中影响面最广的量化加载路径变更。

```
class ModelOptFp4LinearMethod(LinearMethodBase):
    """NVFP4 linear method using the selected FP4 GEMM backend."""

    def __init__(self, quant_config: ModelOptFp4Config):
        # 本方法分发的 FlashInfer FP4 kernel 仅支持 Blackwell。
        # 若不加门禁，加载会在非 Blackwell 卡上先成功，随后在 kernel
        # 内部以不透明的 CUDA 错误失败，用户难以定位到量化配置。
        capability = current_platform.get_device_capability()
        min_capability = quant_config.get_min_capability()
        # 无法读取算力时保持原行为：门禁只负责把崩溃转成可读消息，
        # 不应给原本能工作的卡新增失败路径。
        if capability is not None and capability.to_int() < min_capability:
            raise RuntimeError(
                f"NVFP4 checkpoints need compute capability "
                f"{min_capability // 10}.{min_capability % 10} or newer "
                f"(Blackwell); this GPU is {capability.as_version_str()}. "
                f"Load an FP8 or BF16 checkpoint instead."
            )
        self.quant_config = quant_config
```

python/sglang/multimodal_gen/runtime/pipelines_core/stages/decoding.py

VAE 解码主路径：收窄 enable_tiling 异常守卫为 AttributeError 并显式告警，同时重写两条 OOM 建议，让 --vae-tiling 与 OOM 提示从 " 表象 " 回归 " 本质 "。

```
        # Decode latents
        with autocast_context(vae_dtype, server_args.disable_autocast):
            # 并非所有 VAE 都支持运行时切换 tiling；对不支持的模型明确告警
            # 而非静默丢弃请求，因为下方 OOM 建议会引导用户打开 --vae-tiling。
            if server_args.pipeline_config.vae_tiling:
                try:
                    self.vae.enable_tiling()
                except AttributeError:
                    logger.warning(
                        "--vae-tiling has no effect: %s does not support "
                        "enabling tiling at runtime. Whether it tiles is fixed "
                        "by its VAE config.",
                        type(self.vae).__name__,
                    )
            should_cast_vae = not vae_autocast_enabled
            if not vae_autocast_enabled:
                latents = latents.to(vae_dtype)
            with temporary_module_dtype(
                self.vae, vae_dtype, enabled=should_cast_vae
            ) as vae:
                try:
```

```
                decode_output = self._get_vae_decode_fn(vae, server_args)(latents)
            except Exception as error:
                if "out of memory" in str(error).lower():
                    # decode 在 denoising 之后执行，DiT 与 encoder 虽空闲但仍占用
                    # VRAM；释放它们才是真正杠杆。--vae-cpu-offload 只搬权重，
                    # 对溢出的激活值无济于事，因此不再推荐。
                    if not server_args.pipeline_config.vae_tiling:
                        logger.warning(
                            "OOM detected during VAE decoding. Enable "
                            "--vae-tiling to bound the decode working set, "
                            "and free the components that finished earlier "
                            "with --cpu-offload-components dit,text_encoder."
                        )
                    else:
                        logger.warning(
                            "OOM detected during VAE decoding with tiling enabled. "
                            "Free the components that finished earlier with "
                            "--cpu-offload-components dit,text_encoder, then "
                            "lower the tile size in the model's VAE config, "
                            "then reduce resolution or frame count."
                        )
                raise
```

python/sglang/multimodal_gen/runtime/models/vaes/minimax_h3_video_vae/klvae.py

MiniMax-H3 VAE 是 --vae-tiling 静默失效的典型受害对象，新增 enable_tiling 方法使运行时开关能真正触达这类 VAE，并让新告警逻辑有明确的适配入口。

```
    def enable_tiling(self) -> None:
        """Turn on tiled decode for subsequent decodes.

        `decoder_tiling` 在 `_adaptive_decode` 中每次解码时读取，
        因此此处置位会在下一次调用生效。
        已从 VAE 配置出发做 tiling 的模型（如 MiniMax-H3）不受影响；
        该方法存在的意义是让运行时 `--vae-tiling` 开关能触达本 VAE，
        而不是落入调用方的异常守卫。
        """
        self.decoder_tiling = True

    def _adaptive_decode(self, z):
        # 每次解码都读取 decoder_tiling，因此 enable_tiling() 的
        # 设置会在下一次 decode 调用时立即决定走 tiled 还是整图路径。
        if self.decoder_tiling:
            return self.tiled_decode(z)
        else:
            return self.decode(z)
```

评论区精华

本 PR 没有 reviewer 评论（review_comments_count = 0），唯一的 issue 评论是作者触发 CI 的 "/tag-and-rerun-ci"。作者在 PR body 中记录了三条关键设计论证：其一，"every enable_tiling implementation in the tree takes only optional arguments, so narrowing the guard cannot turn a previously swallowed error into a crash"，即收窄异常守卫前已审计全树全部实现，确保不会引入新崩溃；其二，"A GPU whose capability cannot be read is left alone, so this only ever turns a crash into a message and never blocks a card that would have worked"，即 NVFP4 门禁采取最保守的放行策略；其三，"decode runs after denoising, so the DiT and encoders are idle but may still hold VRAM — freeing them is the actual lever"，澄清了 OOM 建议的归因。

- NVFP4 门禁对无法读取算力的 GPU 是否误伤 (design): 接受该保守策略：无法读取 capability 时保持原有行为，避免新增失败路径。
- --vae-tiling 异常守卫收窄为 AttributeError 的安全性 (correctness): 收窄安全成立，但依赖全树实现的审计结果，未来新增实现需保持同样契约。
- decode OOM 建议的归因与修正 (design): OOM 消息改为分步指引：--cpu-offload-components dit,text_encoder → 降低 tile size → 降低分辨率或帧数。

风险与影响

- 风险：
  1. 异常语义变化：decoding.py 从吞掉所有异常改为只捕获 AttributeError，若未来新增的 enable_tiling 实现抛其他类型的异常，会直接向上传播；作者已审计当前全部 4 个实现，但这是一个依赖全树审计的隐含契约。
  2. Blackwell 真机路径未验证：PR body 明确说明允许路径只在 mocked capability 下执行，真实 B200/B300/5090 上没有跑过 NVFP4 加载；若 flashinfer 的 FP4 kernel 支持范围与 DeviceCapability.to_int() 判断不一致，仍可能存在残余不匹配。
  3. 行为变更影响面：--vae-tiling 从 " 静默无效 " 变为 " 显式警告 "，对依赖旧行为的脚本无功能破坏，但用户可能观察到新的 warning 日志；NVFP4 加载新增 RuntimeError 路径，会改变 pre-Blackwell 机器上的失败时机与报错形式。
  4. 测试对 capability 的强耦合：test_transformer_quant.py 和 test_ideogram4.py 必须 mock get_device_capability 才能通过，说明量化加载路径对运行环境更敏感，CI 环境差异可能带来新的失败模式。 - 影响：用户侧：MiniMax-H3 等 VAE 用户使用 --vae-tiling 时不再被误导，会收到明确的 " 该 VAE 不支持运行时切换 " 警告；decode OOM 时得到的建议从无效的 --vae-cpu-offload 变为可操作的分步指引；pre-Blackwell 用户加载 NVFP4 checkpoint 时会在加载期获得带算力数值与替代方案的可读错误。系统侧：改动局限于 multimodal_gen 的 VAE 解码阶段、MiniMax-H3 VAE 与 NVFP4 量化层，不影响 SRT 推理核心；NVFP4 量化路径的失败时机提前，避免深层 kernel 崩溃造成的排查成本。团队侧：新增的 capability gate 测试为量化加载提供了回归保护，后续新增量化 kernel 或模型时可直接复用。 - 风险标记：VAE tiling 行为语义变更，Blackwell 真机路径未验证，OOM 提示仅消息层变更，guard 缩窄依赖全树审计

关联脉络

- PR #34993 [diffusion] fix: make MiniMax-H3 AdaLN cache rebuild transactional: 同属 MiniMax-H3 模型家族的生命周期管理修复，本 PR 为 MiniMaxH3VideoVAE 补 enable_tiling，延续对该模型运行时健壮性的打磨。
- PR #35339 [diffusion] Per-request lossy accelerations: Cache-DiT, CFG gating, attention backend override: 同属 diffusion 运行时参数与 pipeline 阶段改造，涉及 pipeline_config 与 denoising/decoding 阶段的交互，本 PR 的 vae_tiling 处理与之共享相同的配置面。
- PR #30319 [NPU] Add mxfp4-w4a4 MOE Quantization Support for NPU: 同属量化加载路径，本 PR 在 modelopt NVFP4 路径上新增硬件 capability 门禁，是量化后端硬件适配方向上的补充。
- PR #23112 Add fmha_v2 attention backend for SM90/120: 同属按 GPU 架构能力做功能开关的工程实践，本 PR 的 NVFP4 门禁与 SM90/120 kernel 选择共同反映对 Blackwell 及周边硬件的精细化支持。