PR #25859 完整报告

sgl-project/sglang

[DSA] Make MQA logits free memory ratio configurable

合并时间：2026-05-21 03:27

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25859

执行摘要

- 一句话：将 DSA MQA logits 空闲内存比例改为可配置
- 推荐动作：本 PR 改动较小且逻辑清晰，值得关注的点是环境变量配置的运行时动态性设计（通过静态方法而非类属性获取）以及 NSA→DSA 重命名策略。建议精读 dsa_indexer.py 中的 _get_mqa_logits_budget_bytes 方法，理解预算计算流程。对于维护者，建议补充环境变量的文档说明。

功能与动机

在 H200 这类内存紧张的 GPU 上，DSA MQA logits 的缓存预算过于宽松，容易导致 OOM。通过将空闲内存比例改为可配置并采用更保守的默认值，可以让 logits 路径更早地进行分块处理，避免依赖过时的缓存预算。

实现拆解

1. 新增环境变量：在 python/sglang/srt/environ.py 中增加 SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION = EnvFloat(0.2)，默认值为 0.2，位于 DSA 相关变量组中。
2. 移除硬编码类属性：从 python/sglang/srt/layers/attention/dsa/dsa_indexer.py 的 Indexer 类中删除 _MQA_LOGITS_FREE_MEM_FRACTION = 0.5。
3. 添加动态获取方法：在 Indexer 类中新增静态方法 _mqa_logits_free_mem_fraction()，调用 envs.SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION.get() 实时获取环境变量值。
4. 更新预算计算逻辑：在 _get_mqa_logits_budget_bytes 方法开头调用该静态方法得到 free_mem_fraction，替换原来所有使用 self._MQA_LOGITS_FREE_MEM_FRACTION 的地方，包括静态预算和实时预算计算。
5. 遵循 NSA→DSA 重命名：环境变量名使用 DSA 命名空间，等待 DSA 重命名 PR 合并后落地。

关键文件：

- python/sglang/srt/layers/attention/dsa/dsa_indexer.py（模块 注意力索引器；类别 source；类型 core-logic；符号 _mqa_logits_free_mem_fraction）：核心文件，移除硬编码属性、添加动态获取方法、更新预算计算逻辑

- python/sglang/srt/environ.py（模块 环境配置；类别 source；类型 core-logic；符号 SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION）：新增环境变量声明，定义默认值为 0.2

关键符号：_mqa_logits_free_mem_fraction, _get_mqa_logits_budget_bytes

关键源码片段

python/sglang/srt/layers/attention/dsa/dsa_indexer.py

核心文件，移除硬编码属性、添加动态获取方法、更新预算计算逻辑

```
# python/sglang/srt/layers/attention/dsa/dsa_indexer.py
class Indexer(MultiPlatformOp):
    _MQA_LOGITS_BYTES_PER_ELEM = 4
    _MQA_LOGITS_STATIC_SKIP_ELEMS = 8_000_000
    _MQA_LOGITS_TOTAL_MEM_FRACTION = 0.3
    _mqa_logits_budget_bytes: Dict[int, int] = {}

    @staticmethod
    def _mqa_logits_free_mem_fraction() -> float:
        # 在方法内部动态获取环境变量，支持运行时 override
        return envs.SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION.get()

    def _get_mqa_logits_budget_bytes(self, device_index: int) -> int:
        # 每次调用都获取最新的空闲内存比例
        free_mem_fraction = self._mqa_logits_free_mem_fraction()
        cached_budget = self._mqa_logits_budget_bytes.get(device_index)
        if cached_budget is not None:
            return cached_budget

        total_mem = torch.cuda.get_device_properties(device_index).total_memory
        total_mem_budget = int(total_mem * self._MQA_LOGITS_TOTAL_MEM_FRACTION)
        mem_fraction_static = get_global_server_args().mem_fraction_static
        if mem_fraction_static is None:
            static_budget = total_mem_budget
        else:
            static_free_mem = int(total_mem * max(0.0, 1.0 - mem_fraction_static))
            static_budget = min(
                int(static_free_mem * free_mem_fraction),
                total_mem_budget,
            )
        static_budget = max(1, static_budget)

        if get_is_capture_mode():
            return static_budget

        free_mem, _ = torch.cuda.mem_get_info(device_index)
        # 使用可配置的 free_mem_fraction 替代硬编码 0.5
        budget_bytes = min(int(free_mem * free_mem_fraction), static_budget)
```

```
        budget_bytes = max(1, budget_bytes)
        self._mqa_logits_budget_bytes[device_index] = budget_bytes
        return budget_bytes
```

python/sglang/srt/environ.py

新增环境变量声明，定义默认值为 0.2

```
# python/sglang/srt/environ.py
class Envs:
    # ...
    # DSA Backend (canonical names; fall back to SGLANG_NSA_* with deprecation warning)
    SGLANG_DSA_FUSE_TOPK = EnvBoolWithAlias(True, deprecated_name="SGLANG_NSA_
    FUSE_TOPK")
    SGLANG_DSA_ENABLE_MTP_PRECOMPUTE_METADATA = EnvBoolWithAlias(
        True, deprecated_name="SGLANG_NSA_ENABLE_MTP_PRECOMPUTE_METADATA"
    )
    SGLANG_DSA_PREFILL_DENSE_ATTN_KV_LEN_THRESHOLD = EnvIntWithAlias(
        2048, deprecated_name="SGLANG_NSA_PREFILL_DENSE_ATTN_KV_LEN_THRESHOLD"
    )
    SGLANG_DSA_HIP_DISABLE_PRESHUFFLE = EnvBoolWithAlias(
        False, deprecated_name="SGLANG_NSA_HIP_DISABLE_PRESHUFFLE"
    )
    # 新增：可配置的 MQA logits 空闲内存比例，默认 0.2
    SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION = EnvFloat(0.2)
    SGLANG_USE_FUSED_METADATA_COPY = EnvBool(True)
```

评论区精华

设计争议点：环境变量获取时机

- gemini-code-assist[bot] 指出若在类定义级别直接获取环境变量，会导致值在模块导入后固化，无法通过 override() 或 set() 动态修改（如单元测试或动态重配），建议在方法内部或通过属性方式获取。
- ch-wan 指出环境变量名应遵循 NSA→DSA 重命名（相关 PR #25821），建议使用 DSA 前缀。
- YAMY1234 采纳了重命名建议，并在最终版本中使用了 SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION，同时通过静态方法在运行时获取值而非类定义时获取，解决了动态性顾虑。
- 环境变量获取时机：类定义时 vs 运行时动态获取 (design): 作者采纳建议，通过静态方法 _mqa_logits_free_mem_fraction() 在方法内部调用 envs.SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION.get()，实现运行时动态获取。
- 环境变量命名：遵循 NSA→DSA 重命名 (design): 作者将环境变量名改为 SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION，并确认已处理。

风险与影响

- 风险：

1. 回归风险：默认值从 0.5 改为 0.2，可能影响非 H200 环境下的 MQA logits 预算分配，若原值 0.5 是其他场景的性能调优结果，可能引入性能退化。但 PR body 中的 H200 测试未出现 OOM 且精度指标良好，风险可控。
2. 配置未文档化：环境变量未在文档中说明，用户可能不知晓此配置项。
3. 无测试覆盖：未添加单元测试验证环境变量覆盖和默认值行为，可能存在被回归风险。
   - 影响：影响范围：仅影响 DSA 索引器中 MQA logits 的缓存预算计算逻辑，属于内存管理优化，不涉及模型计算路径变更。 影响程度：中等。对运行 DeepSeek-V3.2 的 H200 节点有直接改善，可降低 OOM 概率；其他 GPU 环境若内存充裕则无明显影响。 用户影响：用户可通过设置 SGLANG_DSA_MQA_LOGITS_FREE_MEM_FRACTION 环境变量自定义空闲内存比例，无其他接口变更。

- 风险标记：默认值变更可能影响非 H200 环境，缺少测试覆盖，配置未文档化

关联脉络

- PR #25821 Rename NSA → DSA: user-facing aliases, file/class/import rename: 本 PR 的环境变量命名和文件路径依赖此重命名 PR，需在其之后合并。