PR #25821 完整报告

sgl-project/sglang

[Refactor] Rename NSA → DSA: user-facing aliases, file/class/import rename

合并时间：2026-05-20 15:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25821

执行摘要

- 一句话：NSA 全面重命名为 DSA，保留向后兼容别名
- 推荐动作：该 PR 是大规模重命名的优秀范例，值得精读学习如何设计向后兼容的别名机制、分步骤迁移、以及使用 git mv 和 shim 文件。重点关注 environ.py 中的 _DeprecatedEnvFallback 混合类和 server_args.py 中的 DeprecatedAliasStoreAction 实现。

功能与动机

PR body 指出 NSA 命名不当，因为该注意力变体是 DeepSeek 特有的，应改名为 DSA (DeepSeek Sparse Attention)。

实现拆解

1. 用户层别名：在 server_args.py 中添加 --dsa-* 规范 CLI 标志，保留 --nsa-* 作为已弃用别名，通过 DeprecatedAliasStoreAction 发出警告。在 environ.py 中引入 _DeprecatedEnvFallback mixin 以及 EnvBoolWithAlias/EnvIntWithAlias，使旧 SGLANG_NSA_* 环境变量在未设置新变量时仍生效。在 attention_registry.py 中将 "dsa" 注册为规范后端键，"nsa" 保持兼容但触发弃用警告。
2. 内部文件 mv：执行 git mv attention/nsa/ → attention/dsa/，nsa_backend.py → dsa_backend.py，以及 jit_kernel/csrc/nsa/ → dsa/，mem_cache/sparsity/algorithms/deepseek_nsa.py → deepseek_dsa.py 等。同时重命名内部文件：nsa_indexer.py → dsa_indexer.py，nsa_backend_mtp_precompute.py → dsa_backend_mtp_precompute.py 等。
3. 类 / 函数 / 变量重命名：将 NativeSparseAttnBackend 改为 DeepseekSparseAttnBackend，NSAMetadata → DSAMetadata，NSAIndexerMetadata → DSAIndexerMetadata，handle_attention_nsa → handle_attention_dsa，is_nsa → is_dsa，nsa_cache_seqlens → dsa_cache_seqlens 等。测试类和方法名同步更新。
4. 导入和引用更新：更新全部 33+ 内部导入位置，包括所有模型文件、调度器、池配置器等。更新文档（docs_new/）和 CI 工作流中的 --nsa-* 为 --dsa-*。
5. 向后兼容 shim：在旧路径 nsa_backend.py 和 nsa/__init__.py 中放置薄重新导出 shim，确保旧导入在新版中继续工作。这些 shim 将在下一个发布版本中移除。

关键文件：

- python/sglang/srt/server_args.py（模块 服务配置；类别 source；类型 configuration；符号 nsa_prefill_backend, dsa_prefill_backend, enable_nsa_prefill_context_parallel, enable_dsa_prefill_context_parallel）：ServerArgs 字段重命名入口，用户 - 服务器配置接口
- python/sglang/srt/environ.py（模块 环境变量；类别 source；类型 dependency-wiring；符号 DeprecatedEnvFallback, EnvBoolWithAlias, EnvIntWithAlias, SGLANG_DSA_FUSE_TOPK）：环境变量别名机制，实现 SGLANG_NSA到 SGLANG_DSA_ 的向后兼容
- python/sglang/srt/layers/attention/attention_registry.py（模块 注册表；类别 source；类型 configuration；符号 AttentionBackendRegistry, dsa, nsa）：注意力后端注册键从 'nsa' 切换到 'dsa'，并保留 'nsa' 为弃用别名
- python/sglang/srt/layers/attention/dsa/dsa_backend_mtp_precompute.py（模块 MTP 预计算；类别 source；类型 dependency-wiring；符号 PrecomputedMetadata, compute_cu_seqlens, DeepseekSparseAttnBackendMTPPrecomputeMixin）：展示类名和字段重命名为 DSA 的核心文件之一
- python/sglang/srt/layers/attention/dsa/dsa_indexer.py（模块 索引器；类别 source；类型 dependency-wiring；符号 BaseIndexerMetadata, get_seqlens_int32, get_page_table_64, get_page_table_1）：索引器核心实现，展示从 nsa 到 dsa 的导入和符号重命名
- python/sglang/srt/layers/attention/nsa_backend.py（模块 后端 shim；类别 source；类型 dependency-wiring；符号 NativeSparseAttnBackend, DeepseekSparseAttnBackend）：向后兼容 shim，从新位置重新导出 NativeSparseAttnBackend

关键符号：handle_attention_dsa, is_dsa, _create_dsa_decode_backend, _create_dsa_prefill_backend, attach_hybrid_dsa_pool_to_hiradix_cache, compute_dsa_seqlens, quantize_k_cache, dequantize_k_cache, transform_index_page_table_prefill, transform_index_page_table_decode

关键源码片段

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

展示类名和字段重命名为 DSA 的核心文件之一

```
from dataclasses import dataclass
from typing import Optional
import torch

@dataclass
class PrecomputedMetadata:
    """Precomputed metadata shared across multiple backend instances."""
    # 基本信息
    cache_seqlens: torch.Tensor # int32, [bs]
    cu_seqlens_k: torch.Tensor # int32, [bs+1]
    # 页表
```

```
    page_indices: torch.Tensor # int32, [bs, max_len]
    real_page_table: Optional[torch.Tensor]
    # DSA 相关字段 —— 命名已从 nsa_* 改为 dsa_*
    seqlens_expanded: torch.Tensor # int32, [expanded_size]
    dsa_cache_seqlens: torch.Tensor # int32, [expanded_size]
    dsa_cu_seqlens_k: torch.Tensor # int32, [expanded_size+1]
    seqlens_expanded_size: int
    max_len: int
    max_seqlen_k: int
    flashmla_metadata: Optional[torch.Tensor] = None

def compute_cu_seqlens(seqlens: torch.Tensor) -> torch.Tensor:
    """Compute cumulative sequence lengths with padding."""
    assert seqlens.dtype == torch.int32
    return torch.nn.functional.pad(
        torch.cumsum(seqlens, dim=0, dtype=torch.int32), (1, 0)
    )
```

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

索引器核心实现，展示从 nsa 到 dsa 的导入和符号重命名

```
# dsa_indexer.py 导入部分 —— 路径和函数名已从 nsa 更新为 dsa
from sglang.jit_kernel.fused_store_index_cache import (
    can_use_dsa_fused_store, # 原 can_use_nsa_fused_store
    fused_store_index_k_cache,
)
from sglang.srt.layers.attention.dsa.utils import (
    aiter_can_use_preshuffle_paged_mqa,
    is_dsa_enable_prefill_cp, # 原 is_nsa_enable_prefill_cp
    is_dsa_prefill_cp_in_seq_split,
)
from sglang.srt.layers.dp_attention import attn_tp_all_gather_into_tensor
from sglang.srt.layers.layernorm import LayerNorm
from sglang.srt.layers.quantization.fp8_kernel import fp8_dtype, is_fp8_fnuz
from sglang.srt.layers.utils import MultiPlatformOp
from sglang.srt.state_capturer.indexer_topk import (
    maybe_capture_indexer_topk,
)
from sglang.srt.utils import (
    add_prefix,
    ceil_align,
    get_bool_env_var,
    is_cuda,
    is_gfx95_supported,
    is_hip,
    is_npu,
)

logger = logging.getLogger(__name__)
```

```
# 多平台检测变量 —— 这些模块级标志后文用于选择索引器实现路径
global _use_multi_stream
_is_cuda = is_cuda()
_is_hip = is_hip()
_is_npu = is_npu()
_use_aiter = get_bool_env_var('SGLANG_USE_AITER') and _is_hip
_is_fp8_fnuz = is_fp8_fnuz()
_is_gfx95_supported = is_gfx95_supported()
# aiter preshuffle 是否可用
_use_aiter_preshuffle = aiter_can_use_preshuffle_paged_mqa()
if _use_aiter and not _use_aiter_preshuffle:
    logger.warning(
        'ROCm DSA indexer: aiter preshuffle paged-MQA path is unavailable '
        '(needs Triton>=3.5.0 or AITER_ENABLE_AOT_GLUON_PA_MQA_LOGITS=1); '
        'falling back to legacy page_size=1 / KVBlockSize=1 path.'
    )
```

评论区精华

PR 无 review 评论；PR body 提醒可能因目录 mv 与上游 PR #23906 (Cuda Graph Runner refactor) 产生 rebase 冲突，建议协调合并顺序。

- 目录 mv 与上游 PR 冲突 (other): PR 已合并，未报告实际冲突

风险与影响

- 风险:
  1. 向后兼容风险：旧 CLI 标志和环境变量别名可能覆盖用户预期，但通过弃用警告和文档引导过渡。
  2. 目录迁移冲突：attention/nsa/ → dsa/ 的 git mv 与同时在修改该目录的上游 PR #23906 存在冲突风险，需协调合并顺序。
  3. 遗漏引用：尽管有 37 次提交和 lint 修复，仍有可能在文档或非关键代码中存在残余 nsa 引用，但已通过多次清理将风险降至最低。 - 影响：影响范围：约 162 个文件、11k 新增 /10k 删除，涉及用户接口（CLI、环境变量、配置文件）、内部 API、导入路径、文档和 CI。用户影响：所有使用 --nsa-* 和 SGLANG_NSA_* 的用户将收到弃用警告，鼓励迁移到 --dsa-* 和 SGLANG_DSA_*；现有配置在当前版本仍兼容。内部影响：开发人员需切换到新的 DSA 命名，旧导入通过 shim 暂时可用。 - 风险标记：向后兼容别名风险，目录迁移冲突风险，跨模块依赖

关联脉络

- PR #23906 Cuda Graph Runner refactor: PR body 指出目录 mv 可能与此 PR 产生 rebase 冲突