# PR #43710 完整报告

vllm-project/vllm

[DSv4] Refactor compressor & Fix ROCm compatibility

合并时间：2026-05-27 10:56

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43710

## 执行摘要

- 一句话：重构 DeepSeek V4 compressor 并修复 ROCm 兼容性
- 推荐动作：建议合并。本次重构显著提升了代码可维护性，并修复了 ROCm 兼容性问题，是向跨平台支持迈出的重要一步。

## 功能与动机

原有的 compressor 实现中，内核分发逻辑分散在 DeepseekCompressor.forward 中，且 _save_partial_states_kernel 直接依赖于 Triton，导致 ROCm 平台无法正确导入 cutedsl 相关内核。本次重构旨在将这些内核移动到合适的平台目录，并简化分发流程，同时修复 ROCm 兼容性问题。

## 实现拆解

1. 移动 _save_partial_states_kernel：从 compressor.py 提取到新的 common/ops/save_partial_states.py，并封装为公共函数 save_partial_states，供所有平台调用。
2. 移动 CUDA 特定内核：将 sparse_attn_compress_cutedsl.py 从 common/ops/ 迁移到 nvidia/ops/，并添加统一的 compress_norm_rope_store_cutedsl 包装函数，根据压缩比率选择融合或分步内核。
3. 重构 Triton 内核分发：在 fused_compress_quant_cache.py 中添加 compress_norm_rope_store_triton 函数，根据 head_dim 和 use_fp4_cache 选择对应的 Triton 内核，消除 compressor.py 中对具体内核的直接引用。
4. 简化 DeepseekCompressor：移除 forward 方法中复杂的条件分支，直接调用 save_partial_states 和平台无关的 compress_norm_rope_store_triton 或 compress_norm_rope_store_cutedsl。
5. 更新模块导出：在 nvidia/ops/__init__.py 和 common/ops/__init__.py 中补齐导出符号，确保导入路径正确。

关键文件：

- vllm/models/deepseek_v4/compressor.py（模块 压缩器；类别 source；类型 data-contract；符号 _save_partial_states_kernel）：核心压缩器类，大幅精简导入和分发逻辑，移除对具体内核的直接引用

- vllm/models/deepseek_v4/nvidia/ops/sparse_attn_compress_cutedsl.py（模块 NVIDIA 内核；类别 infra；类型 rename-or-move；符号 _compress_kv_sparse_attn_cutedsl, compress_kv_sparse_attn_cutedsl, _norm_rope_insert_sparse_attn_cutedsl, norm_rope_insert_sparse_attn_cutedsl）：从 common/ops 迁入，新增统一的 cutedsl 包装函数，按压缩比率分流
- vllm/models/deepseek_v4/common/ops/fused_compress_quant_cache.py（模块 通用内核；类别 infra；类型 infrastructure；符号 _get_sparse_attn_cutedsl_impls, compress_norm_rope_store_triton）：新增 compress_norm_rope_store_triton 函数，统一 Triton 内核分发
- vllm/models/deepseek_v4/common/ops/save_partial_states.py（模块 通用内核；类别 infra；类型 infrastructure；符号 save_partial_states, _save_partial_states_kernel）：新增文件，将从 compressor.py 提取的 _save_partial_states_kernel 封装为公共函数
- vllm/models/deepseek_v4/nvidia/model.py（模块 NVIDIA 模型；类别 source；类型 data-contract）：将 prepare_megamoe_inputs 的导入改为从 nvidia.ops 整体导入，与其他模块一致
- vllm/models/deepseek_v4/nvidia/ops/__init__.py（模块 NVIDIA 内核；类别 infra；类型 infrastructure）：新增导出符号，包括 compress_norm_rope_store_cutedsl 等，形成清晰的公共接口
- vllm/models/deepseek_v4/common/ops/fused_indexer_q.py（模块 通用内核；类别 infra；类型 infrastructure）：更新导入以指向新的 save_partial_states 函数（间接依赖）
- vllm/models/deepseek_v4/common/ops/__init__.py（模块 通用内核；类别 infra；类型 infrastructure）：新增 save_partial_states 的导出，保持模块接口一致
- vllm/models/deepseek_v4/common/ops/cache_utils.py（模块 通用内核；类别 infra；类型 infrastructure）：更新导入以指向新的 save_partial_states

关键符号：save_partial_states, compress_norm_rope_store_triton, compress_norm_rope_store_cutedsl, compress_kv_sparse_attn_cutedsl, norm_rope_insert_sparse_attn_cutedsl, fused_kv_compress_norm_rope_insert_sparse_attn_cutedsl

## 关键源码片段

### vllm/models/deepseek_v4/compressor.py

核心压缩器类，大幅精简导入和分发逻辑，移除对具体内核的直接引用

```
def forward(...):
    # ... state_cache setup ...
    # Save partial states (kv/score split + APE fusion)
    save_partial_states(
        kv, score, ape, positions, state_cache,
        slot_mapping, block_size, state_width, compress_ratio,
    )
    # Compress -> Norm -> RoPE -> Insert
    if head_dim == 512 and self.compress_ratio == 4:
```

```
        # NVIDIA head_dim=512: always use cutedsl fused kernel
        compress_norm_rope_store_cutedsl(...)
    else:
        # Otherwise use Triton kernel (also works on ROCm)
        compress_norm_rope_store_triton(...)
```

### vllm/models/deepseek_v4/common/ops/save_partial_states.py

新增文件，将从 compressor.py 提取的 _save_partial_states_kernel 封装为公共函数

```
@triton.jit
def _save_partial_states_kernel(
    kv_ptr, kv_stride,
    score_ptr, score_stride,
    ape_ptr, ape_stride,
    positions_ptr,
    state_cache_ptr, state_cache_stride0, state_cache_stride1,
    slot_mapping_ptr,
    block_size,
    HEAD_SIZE: tl.constexpr,
    TRITON_BLOCK_SIZE: tl.constexpr,
    STATE_WIDTH: tl.constexpr,
    COMPRESS_RATIO: tl.constexpr,
):
    token_idx = tl.program_id(0)
    slot_id = tl.load(slot_mapping_ptr + token_idx)
    # Skip padded tokens (slot_id == -1)
    if slot_id < 0:
        return
    block_idx = slot_id // block_size
    pos_in_block = slot_id % block_size
    base_ptr = state_cache_ptr + block_idx * state_cache_stride0 + pos_in_block * state_cache_
    stride1

    block = tl.arange(0, TRITON_BLOCK_SIZE)
    mask = block < HEAD_SIZE

    # Load and store kv_state
    kv = tl.load(kv_ptr + token_idx * kv_stride + block, mask=mask)
    tl.store(base_ptr + block, kv, mask=mask)

    # Fused APE: score += ape[position % COMPRESS_RATIO]
    position = tl.load(positions_ptr + token_idx)
    ape_row = position % COMPRESS_RATIO
    ape = tl.load(ape_ptr + ape_row * ape_stride + block, mask=mask)
    score = tl.load(score_ptr + token_idx * score_stride + block, mask=mask)
    tl.store(base_ptr + STATE_WIDTH + block, score + ape, mask=mask)
```

## 评论区精华

审阅者 @zyongye 建议统一内核命名，使 compress_norm_rope_store_triton 和 compress_norm_rope_store_cutedsl 更具区分性。作者 @WoosukKwon 接受建议并更新了函数名称和分发逻辑。

- 统一内核命名 (design): 作者 @WoosukKwon 接受建议，更新了函数名称和分发逻辑，使用 compress_norm_rope_store_triton 和 compress_norm_rope_store_cutedsl 作为统一的入口。

## 风险与影响

- 风险：本次重构未附带专门的测试变更，依赖现有集成测试。ROCm 路径修复属于首次支持，可能在其他 ROCm 配置下仍有未覆盖的问题。此外，统一的分发接口可能引入性能回归，但预期与原行为等价。
- 影响：影响范围：DeepSeek V4 模型的 KV/Score 压缩模块，涉及 NVIDIA 和 AMD ROCm 平台。代码组织更清晰，后续扩展新内核（如自定义压缩比率）更便捷。ROCm 用户将因此次修复获得基本的 compressor 功能支持。
- 风险标记：缺少测试覆盖，ROCm 路径首次修复

## 关联脉络

- PR #43690 [DSv4] Drop _get_compressed_kv_buffer in DeepseekCompressor: 同为 DeepSeek V4 compressor 模块的清理工作，存在代码交接和协作可能。
- PR #43162 [Feat][DSV4] Fuse q pad into deepseek v4 fused kernel: 涉及同一区域（DeepSeek V4 的 kernel 融合），本 PR 重构后的分发接口可与之对接。