PR #47493 完整报告

vllm-project/vllm

[Bugfix] DSV4 TP16 garbage output

合并时间：2026-07-08 12:04

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47493

执行摘要

- 一句话：修复 DSV4 FlashInfer 稀疏 MLA 在打包 KV 布局下的地址错误
- 推荐动作：建议团队精读该 PR，特别是 _remap_flashinfer_index 的设计和 alignment 条件化策略。它展示了如何在不修改核心 kernel 和缓存分配系统的前提下，通过局部数据变换和配置调整兼容底层算子限制。未来当 flashinfer#3856 合入后，应回退这些绕道逻辑。同时，对于 V3.2 等模型使用 FlashInfer 的路径，需额外关注 alignment 变化带来的潜在影响。

功能与动机

关联 issue #47783 报告 DeepSeek-V4 在 TP16 下使用 FLASHINFER_MLA_SPARSE_DSV4 后端输出完全错误（gsm8k ≈ 0%），经二分定位到 #44577（打包 KV 布局）。根本原因是 FlashInfer 稀疏 MLA 解码 kernel 未考虑 packed buffer 的 per-block stride，直接按平坦 token id 索引导致跨块地址越界。该修复需要在保持打包布局（对 KV-transfer 关键）的前提下修正读取路径。

实现拆解

1. 在 vllm/models/deepseek_v4/nvidia/flashinfer_sparse.py 中添加 _packed_block_span(pool: torch.Tensor) -> int 函数，通过 pool.stride(0) // pool.stride(-2) 计算每个物理块包含的 token 数（packed 下大于 block_size）。若不对齐则抛 NotImplementedError。
2. 在 vllm/models/deepseek_v4/common/ops/cache_utils.py 的 build_flashinfer_mixed_sparse_indices 函数中新增 swa_block_span 和 compressed_block_span 参数，并在对应的 Triton kernel _build_flashinfer_mixed_sparse_indices_kernel 内部所有写入稀疏索引的位置调用新定义的设备函数 _remap_flashinfer_index，将每个 slot id 从 block*block_size + offset 变换为 block*block_span + offset。该函数直接内联在 kernel 中，无额外 kernel launch。
3. 在 vllm/models/deepseek_v4/attention.py（DeepseekV4CompressorAttention.get_kv_cache_spec 和 DeepseekV4IndexerCache.get_kv_cache_spec）、vllm/models/deepseek_v4/compressor.py（DeepseekCompressorState.get_kv_cache_spec）、vllm/v1/attention/backends/mla/sparse_swa.py（DeepseekSparseSWACache.get_kv_cache_spec）中，将 alignment 从 576 if

fp8_ds_mla else None 统一改为 576 if fp8_ds_mla else 512。这使得 FlashInfer 后端（非 fp8_ds_mla）的 KV 页对齐到 512B，保证 packed block stride 能被 512 整除，从而 block_span 为整数。同时 Indexer 缓存的对齐也按此条件化，避免对 V3.2 的非预期影响。

4. 测试验证：lm-eval gsm8k 5-shot 在 GB300 TP16 配置下达到 0.9492（匹配基线），现有打包测试 tests/v1/core/test_contiguous_kv_packing.py 8 项全部通过，ruff 和 mypy 检查无新问题。未新增专门测试文件，但通过端到端精度和现有打包测试覆盖。

关键文件：

- vllm/models/deepseek_v4/nvidia/flashinfer_sparse.py（模块 FlashInfer 后端；类别 source；类型 core-logic；符号 _packed_block_span）：核心修复文件：新加 _packed_block_span 函数计算 packed block span，并在构建稀疏索引时传递 block_span 到索引构建函数。
- vllm/models/deepseek_v4/common/ops/cache_utils.py（模块 索引构建；类别 infra；类型 infrastructure；符号 _remap_flashinfer_index）：索引重映射核心：新增 _remap_flashinfer_index 设备函数，在 Triton kernel 中改写 slot id；修改 build_flashinfer_mixed_sparse_indices 接受 block_span 参数并在 kernel 调用中传入。
- vllm/models/deepseek_v4/attention.py（模块 注意力配置；类别 source；类型 data-contract）：修改 FlashInfer 后端的页对齐设置，从 None/576 改为 512（非 fp8_ds_mla 时），确保 packed block stride 可被 token stride 整除；Indexer 缓存对齐也条件化。
- vllm/models/deepseek_v4/compressor.py（模块 压缩器配置；类别 source；类型 data-contract）：调整压缩器状态的页对齐，类似 attention.py，从 None/576 改为 512。
- vllm/v1/attention/backends/mla/sparse_swa.py（模块 稀疏 SWA 层；类别 source；类型 core-logic）：调整稀疏 SWA 层的页对齐，类似其他配置。

关键符号：_packed_block_span, _remap_flashinfer_index, get_kv_cache_spec

关键源码片段

vllm/models/deepseek_v4/nvidia/flashinfer_sparse.py

核心修复文件：新加 _packed_block_span 函数计算 packed block span，并在构建稀疏索引时传递 block_span 到索引构建函数。

```
def _packed_block_span(pool: torch.Tensor) -> int:
    """Per-block stride of ``pool`` in tokens (``stride(0)//stride(-2)``): ==
    block_size for unpacked KV, larger when packed (#44577). Raises if not
    token-aligned."""
    block_stride = pool.stride(0)
    token_stride = pool.stride(-2)
    # 确保 per-block 步长是 per-token 步长的整数倍，否则无法重映射
    if block_stride % token_stride != 0:
        raise NotImplementedError(
            "FLASHINFER_MLA_SPARSE_DSV4 packed KV requires the per-block stride "
            f"({block_stride}) to be a multiple of the per-token stride "
            f"({token_stride}); this layout is not supported yet."
```

```
        )
    return block_stride // token_stride

# 在 _build_sparse_index_metadata 中调用，将 span 传递给索引构建函数
swa_block_span = _packed_block_span(swa_k_cache)
compressed_block_span = _packed_block_span(compressed_kv_cache)
sparse_indices, sparse_topk_lens = build_flashinfer_mixed_sparse_indices(
    ...,
    swa_block_span=swa_block_span,
    compressed_block_span=compressed_block_span,
)
```

vllm/models/deepseek_v4/common/ops/cache_utils.py

索引重映射核心：新增 _remap_flashinfer_index 设备函数，在 Triton kernel 中改写 slot id；修改 build_flashinfer_mixed_sparse_indices 接受 block_span 参数并在 kernel 调用中传入。

```
@triton.jit
def _remap_flashinfer_index(values, block_size, block_span):
    # FlashInfer 的 DSv4 kernel 使用平坦物理 token stride 索引，
    # 因此需要对 packed 页 (#44577) 进行重新映射：
    # 原索引：block * block_size + offset
    # 新索引：block * block_span + offset
    # TODO: remove once flashinfer-ai/flashinfer#3856 is fixed.
    is_valid = values >= 0
    safe_values = tl.where(is_valid, values, 0)
    values = (safe_values // block_size) * block_span
    values += safe_values % block_size
    return tl.where(is_valid, values, -1)

# 在 kernel 内每个写索引位置调用，例如 SWA 索引写入处：
values = _remap_flashinfer_index(values, swa_block_size, swa_block_span)
tl.store(sparse_indices_ptr + token_idx * sparse_indices_stride + offset,
         values,
         mask=mask)
# 压缩器索引写入处类似，使用 compressed_block_span
```

评论区精华

该 PR 无实质 review 评论，但作者在 PR body 中明确比较了三种备选方案：① 禁用打包 KV（#47805）、② 在 core planner 中加 pad 辅助（#47895）、③ 当前方案——复用现有 per-page alignment 字段 + 索引内 remap。作者选择③，因为对 core planner 零侵入、保留打包布局的 KV-transfer 优势，且 alignment 改动局限在 5 个模型本地文件。与 #47895 的唯一差异是 alignment 施加方式：本 PR 通过 MLAAttentionSpec.alignment 字段实现，而非向 kv_cache_utils.py 添加新辅助函数。两个方案都标记为临时工作绕道，等待 flashinfer-ai/flashinfer#3856 上游修复。

- 选择 remap + 页对齐方案而非禁用打包 (design): 采用当前方案，因为它侵入最小且保留打包优势。

风险与影响

- 风险：风险点：① 索引重映射 kernel 正确性：_remap_flashinfer_index 在 Triton kernel 内对所有稀疏索引写位置执行了 (value // block_size) * block_span + value % block_size 变换，逻辑正确但依赖 block_span 在整个 kernel 执行中一致。若 block_span 计算错误（如 pool tensor 的 stride 异常），会导致索引进一步偏差，但 _packed_block_span 的整除性检查提供了防御。② Alignment 条件化影响面：将 alignment 从 None 改为 512 会影响所有非 fp8_ds_mla 的 DeepSeek-V4 后端，包括 bf16 和 e4m3 的 FlashInfer 路径。512B 对齐会略微增加内存占用（约 0.03%），但保证了计算正确性。对于 DeepSeek-V3.2 的 Indexer 路径，原本固定为 576B，现在变更为 576 if fp8_ds_mla else 512，即非 fp8_ds_mla 时从 576 降为 512，这可能导致 V3.2 的非 fp8_ds_mla 页对齐改变，但 PR 未讨论回归。③ 测试覆盖：无新增专门测试，仅依赖端到端精度和打包回归测试。若未来其他模型复用了 FlashInfer 稀疏 MLA 后端且未设置正确 alignment，可能暴露类似问题。
- 影响：影响范围：仅 DeepSeek-V4 用户，且仅使用 FLASHINFER_MLA_SPARSE_DSV4 后端的场景。影响程度：从完全错误的输出恢复至正常精度，对生产环境至关重要。系统层面，打包布局带来的单块 RDMA 传输优化（#44577 初衷）完整保留。团队影响：无，不涉及公共 API 变更。
- 风险标记：索引重映射 kernel 变更，alignment 条件化可能影响 V3.2, 无新增专项测试，Block span 整除性假设

关联脉络

- PR #44577 [DSv4] Pack KV caches into contiguous per-block allocations for DeepSeek V4: 引入打包 KV 布局，由此 PR 修复其与 FlashInfer 稀疏后端的兼容性。
- PR #47783 [Bug]: DSV4 sparse MLA emits garbage (gsm8k ~0%) with the packed KV layout from #44577: 问题报告，直接触发本修复。
- PR #47895 [DSv4] Work around FlashInfer packed KV stride: 同时提出的类似修复，本 PR 采用了其 remap 方法并进行了简化。
- PR #47805 [DSv4] Disable packed KV for FlashInfer backend: 备选方案，本 PR 选择保持打包布局。
- PR #3856 Feature request: honor KV pool stride(0) in trtllm_batch_decode_sparse_mla_dsv4: FlashInfer 上游 issue，预期长期修复后回退本 PR 的绕道。