PR #46275 完整报告

vllm-project/vllm

[ROCm][Perf][DSV4] Enable split sparse decode on gfx942

合并时间：2026-07-16 20:28

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46275

执行摘要

- 一句话：启用 gfx942 split sparse decode，解码性能提升 40%
- 推荐动作：值得合并。性能提升明显，且风险极低。评审已经通过。建议后续关注是否有其他架构（如 gfx940）也能受益，但需单独验证。

功能与动机

DeepSeek-V4 稀疏 MLA 解码在 gfx942 上原本使用 fallback monolithic kernel，即使 split partial/reduce 路径在 gfx942 上也能正确运行且性能更优。此 PR 将该路径的硬件守卫从仅 gfx950 扩展到 gfx942，以提升 gfx942 上的解码性能。详见 PR 正文："The latest MI300X/gfx942 profiles show the sparse decode work itself drops by about 40% with the split path."

实现拆解

1. 扩展硬件守卫：在 vllm/v1/attention/ops/rocm_aiter_mla_sparse.py 中，将条件从 if not _ON_GFX950 改为 if not (_ON_GFX942 or _ON_GFX950)，使 gfx942 也能走 split decode 路径，否则 fallback 到原 monolithic kernel。
2. 更新测试辅助函数：在 tests/kernels/attention/test_rocm_triton_attn_dsv4.py 中，将原有的 _on_gfx950() 重命名为 _on_split_decode_arch()，并同时检查 _ON_GFX942 和 _ON_GFX950，使测试在 gfx942 和 gfx950 上都能运行。相应地将 pytest skipif 条件从 requires_gfx950 重命名为 requires_split_decode_arch。
3. 重构测试中的量化缓存函数：将 _pack_fp8_ds_mla_cache 和 _read_fp8_ds_mla_cache 的参数从 is_extra 改为 use_fnuz，并使用 vllm.models.deepseek_v4.common.ops.cache_utils.quantize_and_insert_k_cache 替换手动循环，使测试与生产代码的量化逻辑对齐。

关键文件：

- vllm/v1/attention/ops/rocm_aiter_mla_sparse.py（模块 注意力算子；类别 source；类型 core-logic；符号 _rocm_sparse_attn_decode_ragged_triton）：核心变更：修改 split decode 路径的硬件守卫条件，使 gfx942 也能使用优化后的 split partial/reduce kernel，而不再 fallback 到 monolithic kernel。这是性能提升的直接原因。
- tests/kernels/attention/test_rocm_triton_attn_dsv4.py（模块 测试；类别 test；类型 test-coverage；符号 _on_split_decode_arch, requires_split_decode_arch, _pack_fp8_ds_mla_cache, _read_fp8_ds_mla_cache）：测试覆盖更新：将硬件检测函数

从仅 gfx950 扩展为 gfx942/gfx950，并重构了量化缓存辅助函数以对齐生产代码。确保 split decode 路径在 gfx942 上被测试覆盖。

关键符号：_rocm_sparse_attn_decode_ragged_triton, _on_split_decode_arch, requires_split_decode_arch, _pack_fp8_ds_mla_cache, _read_fp8_ds_mla_cache

关键源码片段

vllm/v1/attention/ops/rocm_aiter_mla_sparse.py

核心变更：修改 split decode 路径的硬件守卫条件，使 gfx942 也能使用优化后的 split partial/reduce kernel，而不再 fallback 到 monolithic kernel。这是性能提升的直接原因。

```
# vllm/v1/attention/ops/rocm_aiter_mla_sparse.py
# 在函数 _rocm_sparse_attn_decode_ragged_triton 中，
# 决定使用哪个 kernel 的关键守卫条件：

if not (_ON_GFX942 or _ON_GFX950): # 修改前：if not _ON_GFX950
    # Fallback path for un-tuned architectures.
    # 对于 gfx942/gfx950 之外的架构，使用 monolithic ragged kernel。
    block_k = 16 if head_dim >= 256 else 32
    _sparse_attn_decode_ragged_kernel[(num_queries, heads_blocks)](
        q, q_stride_q, q_stride_h, ...
    )
else:
    # Split partial + reduce path (tuned for gfx942/gfx950)
    _sparse_attn_decode_partial_kernel[(num_queries, heads_blocks)](...)
    _sparse_attn_decode_reduce_kernel[(num_queries, heads_blocks)](...)
```

tests/kernels/attention/test_rocm_triton_attn_dsv4.py

测试覆盖更新：将硬件检测函数从仅 gfx950 扩展为 gfx942/gfx950，并重构了量化缓存辅助函数以对齐生产代码。确保 split decode 路径在 gfx942 上被测试覆盖。

```
# tests/kernels/attention/test_rocm_triton_attn_dsv4.py
# 硬件架构检测函数：从仅检查 gfx950 扩展为同时检查 gfx942 和 gfx950

def _on_split_decode_arch() -> bool: # 原函数名 _on_gfx950
    if not current_platform.is_rocm():
        return False
    try:
        from vllm.platforms.rocm import _ON_GFX942, _ON_GFX950
        return bool(_ON_GFX942 or _ON_GFX950) # 原返回值：bool(_ON_GFX950)
    except Exception:
        return False

# 对应的 pytest skipif 标记
requires_split_decode_arch = pytest.mark.skipif( # 原名 requires_gfx950
    not _on_split_decode_arch(),
    reason="split-K decode kernel is only tuned for AMD gfx942/gfx950",
)
```

```
# 同时，量化缓存辅助函数从手动循环改为调用生产代码的量化插入函数，
# 参数从 is_extra 改为 use_fnuz，与生产代码保持一致。
def _pack_fp8_ds_mla_cache(
    kv: torch.Tensor, block_size: int, use_fnuz: bool # 原 : is_extra: bool = False
) -> torch.Tensor:
    ...
    from vllm.models.deepseek_v4.common.ops.cache_utils import (
        quantize_and_insert_k_cache,
    )
    slot_mapping = torch.arange(num_tokens, dtype=torch.int64, device=kv.device)
    quantize_and_insert_k_cache(
        kv, cache, slot_mapping,
        block_size=block_size, use_fnuz=use_fnuz,
    )
    return cache
```

评论区精华

评审人 tjtanaa 在 Approval 中表示："LGTM. Thanks for fixing the extra and fnuz semantics. It is clearer than the one introduced in https://github.com/vllm-project/vllm/pull/46080 。" 说明之前 PR#46080 中引入的语义不够清晰，此 PR 做了清理。

- 审批意见：感谢修复 extra 和 fnuz 语义 (other): 评审认可，认为语义比之前的 PR 更清晰。

风险与影响

- 风险：风险较低。核心变更仅一行条件判断的修改，且经过测试验证（41 个测试通过）、正确性校验（deterministic smoke request 返回 42、GSM8K 精度一致）和端到端 benchmark 对比（性能提升无退化）。需要注意的是：
  - 此优化仅适用于 gfx942/gfx950，其他 AMD 架构仍使用 fallback 路径，不影响。
  - 如果将来有其他架构也能走 split 路径，需要更新守卫。
  - 测试中重构了量化缓存函数，但已通过正确性 diff 验证与生产逻辑一致。
  - 影响：对使用 DeepSeek-V4 模型在 AMD gfx942 上的用户：解码性能显著提升（kernel 级别约 40%，端到端 decode-heavy TPOT 降低 4.5%，高吞吐 TTFT 降低 13%），且无需任何配置变更。对其他用户无影响，因为守卫条件确保了只有 gfx942/gfx950 受影响。团队维护成本低，因为变更集中且简单。
  - 风险标记：GPU 架构特定，依赖 ROCm 平台，测试覆盖关键路径

关联脉络

- PR #46080 [ROCm][DSV4] 之前引入的量化缓存语义不够清晰：评审人提到此 PR 修复了 PR#46080 中引入的 extra 和 fnuz 语义问题，使其更清晰。相关文件同为测试文件中的量化缓存函数。