PR #51216 完整报告

vllm-project/vllm

[ROCm][AMD] Enable preshuffled sparse indexing for 16-token blocks

合并时间：2026-08-15 01:38

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51216

执行摘要

- 一句话：ROCm MLA 稀疏索引器启用 16 token 块 preshuffled 路径
- 推荐动作：值得快速精读。虽然只有 16 行改动，但展示了用 MultipleOf 表达 kernel 能力约束、让块大小协商更灵活的设计思路，以及用生产真实后端类编写回归测试防回退的实践，对后续维护 ROCm MLA 稀疏路径的工程师有参考价值。

功能与动机

PR body 明确说明：此前 --block-size 16 会选中 kernel block size 1，从而禁用 AITER 的 preshuffled FP8 paged-MQA 路径；而 preshuffled 变体性能明显优于非 shuffled 变体，“we were leaving some performance on the table”。本 PR 的目标是让块大小协商在 ROCm 上保留 16 对齐的块，从而走回更快的 preshuffled 内核。

实现拆解

1. 能力声明修改（核心逻辑）：vllm/v1/attention/backends/mla/indexer.py 中 DeepseekV32IndexerBackend.get_supported_kernel_block_sizes() 由 [1, 64]（ROCm 分支）改为 [1, MultipleOf(16)]，NVIDIA 分支保持 [64] 不变；vllm/v1/attention/backends/mla/rocm_aiter_mla_sparse.py 中 ROCMAiterMLASparseBackend.get_supported_kernel_block_sizes() 由 [1, 64] 改为 [1, MultipleOf(16)]。MultipleOf(16) 表达“任意 16 的倍数都合法”，使 select_common_block_size 在协商时能保留 16 而不会退化到 1。
2. 新增回归测试：tests/v1/worker/test_gpu_model_runner.py 新增 test_select_common_block_size_accepts_rocm_sparse_block_size_16，用 monkeypatch 把 current_platform.is_rocm 置为 True，并使用生产环境的真实后端类（而非 mock）验证 select_common_block_size(16, [DeepseekV32IndexerBackend, ROCMAiterMLASparseBackend]) == 16，防止任一后端后续收紧能力导致静默回退。
3. 验证与配套：本地 pytest tests/v1/worker/test_gpu_model_runner.py -k select_common_block_size 4 个用例全部通过，ruff 等 pre-commit 检查全部通过；不涉及配置、文档、schema 或部署配套改动。性能验证在 4×MI355X、GLM-5.2 MXFP4、TP4、并发 4、block size 16、50k 输入 /1k 输出负载下完成。

关键文件：

- vllm/v1/attention/backends/mla/indexer.py（模块 索引器；类别 source；类型 core-logic；符号 DeepseekV32IndexerBackend.get_supported_kernel_block_sizes）：DeepseekV32IndexerBackend 的能力声明从固定 64 放宽为 MultipleOf(16)，是让 block size 16 走 preshuffled 路径的关键入口之一，且保留 NVIDIA 分支不变。
- vllm/v1/attention/backends/mla/rocm_aiter_mla_sparse.py（模块 稀疏后端；类别 source；类型 core-logic；符号 ROCMAiterMLASparseBackend.get_supported_kernel_block_sizes）：ROCMAiterMLASparseBackend 同步放宽块大小能力，与 indexer 对齐，确保 AITER preshuffled FP8 paged-MQA 内核在 16 对齐块下可用。
- tests/v1/worker/test_gpu_model_runner.py（模块 模型运行器；类别 test；类型 test-coverage；符号 test_select_common_block_size_accepts_rocm_sparse_block_size_16）：新增针对真实后端的回归测试，验证 block size 16 在 ROCm 下能被两个稀疏索引器后端共同接受，是防止未来回退的主要防线。

关键符号：DeepseekV32IndexerBackend.get_supported_kernel_block_sizes, ROCMAiterMLASparseBackend.get_supported_kernel_block_sizes, test_select_common_block_size_accepts_rocm_sparse_block_size_16

关键源码片段

vllm/v1/attention/backends/mla/indexer.py

DeepseekV32IndexerBackend 的能力声明从固定 64 放宽为 MultipleOf(16)，是让 block size 16 走 preshuffled 路径的关键入口之一，且保留 NVIDIA 分支不变。

```
class DeepseekV32IndexerBackend(AttentionBackend):
    """DeepSeek V3.2 稀疏索引器后端。"""

    @staticmethod
    def get_supported_kernel_block_sizes() -> list[int | MultipleOf]:
        # ROCm 上声明支持 1 或任意 16 对齐的块大小：
        # 这样 `--block-size 16` 时也能选中 preshuffled 稀疏解码路径，
        # 而不是回退到 kernel block size 1 的非 shuffled 版本。
        # 非 ROCm 平台保持只支持 64，避免改变 NVIDIA 行为。
        return [1, MultipleOf(16)] if current_platform.is_rocm() else [64]
```

vllm/v1/attention/backends/mla/rocm_aiter_mla_sparse.py

ROCMAiterMLASparseBackend 同步放宽块大小能力，与 indexer 对齐，确保 AITER preshuffled FP8 paged-MQA 内核在 16 对齐块下可用。

```
class ROCMAiterMLASparseBackend(AttentionBackend):
    # 支持 FP8 与 BF16 等 KV cache dtype 的 ROCm AITER 稀疏 MLA 后端
    supported_dtypes: ClassVar[list[torch.dtype]] = [torch.float16, torch.bfloat16]
    supported_kv_cache_dtypes: ClassVar[list[CacheDType]] = [
        "auto", "float16", "bfloat16", "fp8", "fp8_e4m3",
    ]

    @staticmethod
    def get_supported_kernel_block_sizes() -> list[int | MultipleOf]:
```

```
        # 与 DeepseekV32IndexerBackend 同步放开：任意 16 对齐的块大小
        # 均可走 preshuffled FP8 paged-MQA 内核，覆盖 `--block-size 16`
        # 场景，避免性能回退到非 preshuffled 变体。
        return [1, MultipleOf(16)]
```

tests/v1/worker/test_gpu_model_runner.py

新增针对真实后端的回归测试，验证 block size 16 在 ROCm 下能被两个稀疏索引器后端共同接受，是防止未来回退的主要防线。

```
def test_select_common_block_size_accepts_rocm_sparse_block_size_16(monkeypatch):
    # 模拟 ROCm 平台，并用生产环境真实的后端类（而非 mock）验证
    # 16 token 块大小能被两个稀疏索引器后端同时接受；
    # 防止后续某个后端收紧能力时静默回退到慢速路径。
    monkeypatch.setattr(current_platform, "is_rocm", lambda: True)

    selected_size = select_common_block_size(
        16,
        [DeepseekV32IndexerBackend, ROCMAiterMLASparseBackend],
    )
    assert selected_size == 16
```

评论区精华

没有实质性的技术讨论。claude[bot] 仅提示该 PR 来自 fork，自动 review 被禁用，需要 maintainer 手动触发；AndreasKaratzas 直接触发 /ci run 并 approve 后合入，未提出任何代码层面的疑问或修改要求。

- fork PR 的自动 review 与快速合入 (other): 没有未解决的讨论；审核人以 approve 表示变更符合预期并合入。

风险与影响

- 风险：
  1. 中间块大小未覆盖：MultipleOf(16) 同时放开了 32、48、80 等大小，但新增测试只覆盖 16。若 AITER preshuffled 内核或 indexer 的 chunk 逻辑在非 16/64 大小上存在边界问题，当前测试无法发现，建议后续补充一个 32 的用例。
  2. 平台隔离充分：indexer.py 的改动位于 is_rocm() 分支内，rocm_aiter_mla_sparse.py 本身是 ROCm 专属后端，因此 NVIDIA 等其它平台行为完全不变，回归风险低。
  3. 性能影响正向：block size 16 用户可获得约 3.64 倍的内核加速，端到端吞吐提升约 2.5%，且 preshuffled 只是布局重排而非近似内核，无精度变化风险。 - 影响：影响范围集中在 ROCm + MLA + 稀疏索引器（如 GLM-5.2 等 MLA 模型）且使用 --block-size 16 的用户，这类场景的内核耗时显著下降，端到端延迟与吞吐有约 2.5%~3% 的改善。对 CUDA 路径无影响，对非 16 对齐的块大小行为不变。团队维护成本极低：改动集中、语义清晰，并有回归测试作为防回退防线。 - 风险标记：中间块大小未测试，平台特有条件分支，回归保护依赖单一测试

关联脉络

- PR #52164 [Attention][DSA] Take the native decode path for MTP=3 on SM90: 同样修改 vllm/v1/attention/backends/mla/indexer.py，在 SM90 上为 MTP 场景扩展 indexer 能力，与本 PR 在 ROCm 侧的能力扩展属于同一索引器后端的持续演进。
- PR #51704 [5/N][KV-Cache Layout Refactor] Backend-published KV packing via customize_spec: 把后端能力以规格形式发布给上层，与本 PR 通过 get_supported_kernel_block_sizes 声明块大小约束的思路一致，都是将“后端能做什么”显式暴露给选择器。