PR #47017 完整报告

vllm-project/vllm

[ROCm] Enable DeepSeek-V4 on gfx11

合并时间：2026-08-12 08:55

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47017

执行摘要

- 一句话：在 ROCm gfx11 设备上启用 DeepSeek-V4 检查点支持
- 推荐动作：值得精读。改动虽只有 24 行，但体现了 " 最小变更解锁平台能力 " 的设计方式：通过 op 级 fallback 而非整体放宽平台能力判断来复用既有 AITER 稀疏索引器路径；同时 review 中关于保留原始守卫的坚持值得借鉴，可关注该模式在后续 ROCm 平台适配 PR 中的延续。

功能与动机

PR body 明确说明目标："This PR enables DeepSeek-V4 checkpoints on ROCm gfx11/RDNA devices"，通过 "removes Python-side blockers in the ROCm sparse-indexer path" 并 "allows DeepSeek-V4 checkpoints mapped to INCConfig to pass ROCm platform validation"。验证检查点为 Intel/DeepSeek-V4-Flash-W4A16-AutoRound，即在没有启用 AITER 的 RDNA 设备上无法加载该检查点的问题。

实现拆解

变更入口：三个文件协同，核心在 vllm/_aiter_ops.py 的 register_ops_once() 与 vllm/model_executor/layers/sparse_attn_indexer.py 的 SparseAttnIndexer.forward_hip()，配套平台校验配置在 vllm/platforms/rocm.py。

1. 算子注册路径拆分为 gfx11 专属分支（vllm/_aiter_ops.py）：原逻辑在 is_aiter_found_and_supported() or is_aiter_found_and_supported_on_rdna4() 不成立时直接 return。本 PR 改为：非 AITER 且平台为 ROCm 时，若 on_gfx11() 成立则只注册 rocm_aiter_sparse_attn_indexer（声明 mutates_args=["topk_indices_buffer"] 与 fake 实现），并置 _OPS_REGISTERED 后 return；否则保持原 return 行为。这样既解锁 gfx11，又不影响其他 ROCm 平台在无 AITER 时的原始行为，AITER 全量注册主路径保持不变。
2. forward_hip 条件放宽（vllm/model_executor/layers/sparse_attn_indexer.py）：在原有 rocm_aiter_ops.is_enabled() or rocm_aiter_ops.is_rdna_aiter_enabled() 之上追加 or on_gfx11()，使 gfx11 命中自定义算子调用路径；非 gfx11 且无 AITER 时仍抛出原有 RuntimeError 错误提示。
3. 平台量化校验放行（vllm/platforms/rocm.py）：在 RocmPlatform.supported_quantization 列表 "fbgemm_fp8" 与 "quark" 之间插入 "inc"，

使 Intel/DeepSeek-V4-Flash-W4A16-AutoRound 这类 INC 量化检查点通过平台验证。

4. 测试与部署配套：本 PR 未附带自动化测试，验证完全依赖手动流程（gfx1151/gfx1100 实机）；部署侧需注意 gfx1100 下需设置 AMDGCN_USE_BUFFER_OPS=0 规避 Triton 3.6.0 AMD 后端对 fused_q_kv_rmsnorm Triton 内核的编译失败。

关键文件：

- vllm/_aiter_ops.py（模块 算子注册；类别 source；类型 dependency-wiring；符号 register_ops_once）：核心变更文件。在 register_ops_once() 中为 gfx11 非 AITER 平台单独注册 rocm_aiter_sparse_attn_indexer 算子，并保留原 AITER 全量注册路径与守卫逻辑。
- vllm/model_executor/layers/sparse_attn_indexer.py（模块 稀疏注意力；类别 source；类型 data-contract；符号 SparseAttnIndexer.forward_hip）：关键调用路径。SparseAttnIndexer.forward_hip() 在原有 AITER 条件上追加 on_gfx11()，使 gfx11 设备可调用自定义算子，同时保留其他平台的错误提示。
- vllm/platforms/rocm.py（模块 平台层；类别 source；类型 configuration；符号 RocmPlatform.supported_quantization）：平台配置变更。RocmPlatform.supported_quantization 加入 inc，使 INC 量化检查点通过 ROCm 平台验证，是 DeepSeek-V4 检查点可加载的前置条件。

关键符号：register_ops_once, SparseAttnIndexer.forward_hip

关键源码片段

vllm/_aiter_ops.py

核心变更文件。在 register_ops_once() 中为 gfx11 非 AITER 平台单独注册 rocm_aiter_sparse_attn_indexer 算子，并保留原 AITER 全量注册路径与守卫逻辑。

```
@staticmethod
def register_ops_once() -> None:
    global _OPS_REGISTERED

    # 主路径：AITER 已安装且受支持（含 RDNA4）时，走下方全量注册
    if not (
        is_aiter_found_and_supported() or is_aiter_found_and_supported_on_rdna4()
    ):
        # 非 AITER 场景下，仅对 ROCm gfx11 单独注册稀疏注意力索引器算子，
        # 这是 DeepSeek-V4 在 RDNA 设备上运行所依赖的路径；其他 ROCm
        # 平台不注册任何 AITER 算子，保持原有行为
        if not current_platform.is_rocm():
            return

        from vllm.platforms.rocm import on_gfx11

        if on_gfx11() and not _OPS_REGISTERED:
            direct_register_custom_op(
                op_name="rocm_aiter_sparse_attn_indexer",
                op_func=rocm_aiter_sparse_attn_indexer,
```

```
            # 该算子会原地改写 topk_indices_buffer，必须声明 mutates 参数
            mutates_args=["topk_indices_buffer"],
            fake_impl=rocm_aiter_sparse_attn_indexer_fake,
            dispatch_key=current_platform.dispatch_key,
        )
        _OPS_REGISTERED = True
    return

if not _OPS_REGISTERED:
    # 原有 AITER 全量注册逻辑（asm_moe、fused_moe、topk 等）保持不变，
    # 此处省略后续 direct_register_custom_op 调用
    ...
```

vllm/model_executor/layers/sparse_attn_indexer.py

关键调用路径。SparseAttnIndexer.forward_hip() 在原有 AITER 条件上追加 on_gfx11()，使 gfx11 设备可调用自定义算子，同时保留其他平台的错误提示。

```
def forward_hip(
    self,
    hidden_states: torch.Tensor,
    q_quant: torch.Tensor | tuple[torch.Tensor, torch.Tensor],
    k: torch.Tensor,
    weights: torch.Tensor,
):
    assert not self.use_fp4_cache, "AMD platform doesn't support fp4 cache yet"
    assert isinstance(q_quant, torch.Tensor), (
        "AMD sparse_attn_indexer expects a single FP8 q_quant tensor"
    )
    from vllm.platforms.rocm import on_gfx11

    # 在 AITER 可用或 gfx11 设备上调用自定义稀疏注意力索引器算子；
    # 其他 ROCm 平台若未启用 AITER，仍走下方原始错误路径
    if (
        rocm_aiter_ops.is_enabled()
        or rocm_aiter_ops.is_rdna_aiter_enabled()
        or on_gfx11()
    ):
        return torch.ops.vllm.rocm_aiter_sparse_attn_indexer(
            hidden_states,
            _encode_layer_name(self.k_cache.prefix),
            self.k_cache.kv_cache,
            q_quant,
            k,
            weights,
            self.quant_block_size,
            self.scale_fmt,
            self.topk_tokens,
            self.head_dim,
            self.max_model_len,
```

```
            self.max_total_seq_len,
            self.topk_indices_buffer,
            skip_k_cache_insert=self.skip_k_cache_insert,
        )
    raise RuntimeError(
        "Sparse attention indexer ROCm path is only supported on AITER. "
        "Please enable aiter with VLLM_ROCM_USE_AITER=1"
    )
```

评论区精华

核心交锋围绕平台守卫的取舍：shen-shanshan 在 CHANGES_REQUESTED 中指出不应直接删除 AITER 检测，至少加一个独立的 if _ON_GFX11 块；作者据此恢复原始 guard，gfx11 单独注册 rocm_aiter_sparse_attn_indexer，并保留其他非 AITER ROCm 平台的错误路径，最终获得 APPROVED（"Overall LGTM."）。此外 skyguan92 在 issue 评论中给出独立的 gfx1100 物理机验证，但明确说明是 exact feature-patch replay 而非当前 PR 头的完整树构建，且未覆盖独立的 inc 量化追加改动。

- register_ops_once 不应直接删除原始 AITER 守卫 (design): 作者恢复原始 AITER guard，gfx11 单独注册 rocm_aiter_sparse_attn_indexer，非 gfx11 非 AITER 平台保持原有返回行为。review 最终 APPROVED。
- forward_hip 同样需保留原始错误路径 (design): forward_hip 显式允许 on_gfx11()，同时保留其他非 AITER ROCm 平台的原始错误提示。

风险与影响

- 风险：
  1. 平台校验全局放宽：supported_quantization 是类级列表，对所有 ROCm 平台生效，非 gfx11 设备也可能接受 INC 检查点并加载到后续不支持的路径上（如非 AITER 平台触发 forward_hip 的 RuntimeError）。
  2. 运行时依赖算子注册顺序：forward_hip() 在 gfx11 上调用 torch.ops.vllm.rocm_aiter_sparse_attn_indexer，要求 register_ops_once() 先执行成功；若注册被跳过，将出现找不到算子的运行时错误。
  3. 缺少自动化测试覆盖：PR 未带单元或集成测试，回归风险完全依赖人工验证；register_ops_once 的控制流改动影响所有平台。
  4. Triton 编译 workaround 依赖：gfx1100 验证依赖 AMDGCN_USE_BUFFER_OPS=0 规避 Triton 3.6.0 AMD 后端编译失败，该问题未在本 PR 中根治，后续 Triton 升级可能改变行为。 - 影响：用户影响：ROCm gfx11/RDNA 用户在未安装 AITER 的情况下也能加载 DeepSeek-V4 AutoRound 检查点并完成服务部署，是新的硬件 - 模型组合解锁。系统影响：算子注册入口和稀疏注意力索引器调用路径的平台 gating 逻辑被细分，其他 ROCm 平台保持原行为。团队影响：平台校验白名单纳入 inc 量化，后续 DeepSeek-V4 相关检查点在 ROCm 上的验证流程得以简化。整体影响面窄，改动量小。 - 风险标记：缺少自动化测试覆盖，平台校验全局放宽，依赖 Triton 编译 workaround，运行时依赖算子注册顺序

关联脉络

- PR #51838 [Refactor] Delete dead code in models: 涉及 vllm/models/deepseek_v4/amd/model.py 等 DeepSeek-V4 模型文件清理，与本次 DeepSeek-V4 ROCm 支持同属一条功能线。
- PR #49758 [ROCm][MoE] Fix expert_map vs AITER expert_mask for non-AITER experts under EP: 修复 ROCm 上非 AITER MoE 内核的 mask 解读问题，与本 PR 的 AITER 算子注册 / 调用约定直接相关。
- PR #48223 [Perf][ROCm] Dual-stream decode with hipgraphs: ROCm 上共享专家双流解码优化，同属 ROCm/AITER 路径的持续演进。