PR #48886 完整报告

vllm-project/vllm

[ROCm] [BugFix] Fix Quark GLM-5.2 Checkpoint inference: indexer wk per-channel FP8 dequant + missing sparse-MLA metadata fields

合并时间：2026-07-28 06:38

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48886

执行摘要

- 一句话：修复 Quark GLM-5.2 加载与 ROCm 稀疏 MLA 元数据崩溃
- 推荐动作：值得精读。这是“共享 forward 重构遗漏后端同步”的典型修复，supports_dense_mha_prefill 能力标志模式可以迁移到其他能力检测场景；同时展示了多贡献者合并重复 PR、保留回归测试归属的协作流程。建议关注后续 ROCm aiter sparse MLA 的 dense-MHA prefill follow-up，以及 #49275 对 NVFP4 MTP 加载问题的补充修复。

功能与动机

PR body 明确指出两个独立 bug 阻塞 quark quantized GLM-5.2 checkpoints（Attn quantized to PTPC FP8）在 ROCm（MI355X / gfx950）端到端运行：加载期崩溃是 IndexError: tuple index out of range（deepseek_v2.py 的 block_size = weight_fp8.shape[1] // scale_inv.shape[1]），图捕获期崩溃是 AttributeError: 'ROCMAiterMLASparseMetadata' object has no attribute 'num_decodes' 与 'prefill_max_seq_len'。jimmy-adams 在评论中独立确认在 AMD MI350（gfx950）上用 GLM-5.1-NVFP4 + MTP 也遇到同一 crash，证明该问题影响面超出 GLM-5.2 单一模型。

实现拆解

1. FP8 indexer wk 的 per-channel scale 反量化（vllm/model_executor/models/deepseek_v2.py）：_try_load_fp8_indexer_wk 在同时拿到 FP8 权重与 weight_scale_inv 后，原先无条件按 weight_fp8.shape[1] // scale_inv.shape[1] 推算 block size 并构造 GroupShape(block_size, block_size)。修复改为先检查 scale_inv.ndim：1-D 时构造 GroupShape(1, weight_fp8.shape[1])（per-output-channel），2-D 时走原 block-wise 路径。改动仅作用于加载期反量化，不改变 fused wk_weights_proj 的后续计算语义。
2. 引入 supports_dense_mha_prefill 能力标志（vllm/v1/attention/backend.py + vllm/model_executor/layers/attention/mla_attention.py）：AttentionImplBase 新增 ClassVar[bool] = True 默认值；mla_attention.py 的初始化逻辑改为先判断 self.impl.is_sparse and not self.impl.supports_dense_mha_prefill，命中则跳过 get_mla_prefill_backend 直接置 prefill_backend = None，从根因上避免为无 forward_mha 的 impl 初始化 dense-MHA prefill backend。原有 try/except 分支保留，用于处理“无任何后端支持但用户显式配置”时的报错语义，避免静默降级。

3. 补齐 ROCMAiterMLASparseMetadata 的 split 字段（vllm/v1/attention/backends/mla/rocm_aiter_mla_sparse.py）：dataclass 增加 num_decodes、num_prefills、num_decode_tokens、prefill_max_seq_len、prefill 默认字段；build() 入口复用既有工具函数 split_decodes_and_prefills（decode_threshold 取 self.reorder_batch_threshold or 1）计算三个计数并传入 metadata 构造；ROCMAiterMLASparseImpl 声明 supports_dense_mha_prefill = False，保证共享 forward 始终走 MQA-only 路径。
4. 回归测试（tests/kernels/attention/test_rocm_aiter_mla_sparse_metadata_sync.py）：从 #48722 cherry-pick 约 75 行测试，新增 _make_mixed_common_metadata 与 _patch_build_deps（stub 掉 aiter 内核、triton 辅助函数和 CUDA stream，使 build() 可在 CPU 上运行），并新增两个用例：decode-only batch 断言 num_decodes == 2、num_prefills == 0、num_decode_tokens == 2；decode+prefill 混合 batch 断言 num_decodes == 1、num_prefills == 1、num_decode_tokens == 1，同时确认 prefill_max_seq_len == 0 且 prefill is None（MQA-only 语义）。
5. 验证：4x MI355X（gfx950）上以 lm_eval 运行 Quark 量化 GLM-5.2 的 gsm8k（5-shot），flexible-extract 与 strict-match 均为 0.9295。

关键文件：

- vllm/model_executor/layers/attention/mla_attention.py（模块 注意力层；类别 source；类型 data-contract）：共享 MLA forward 的核心文件。#47327 重构后这里新增了从 metadata 读取 split 字段的依赖，本 PR 在 prefill_backend 初始化处引入 supports_dense_mha_prefill 能力分支，从根因上避免无 dense-MHA 的稀疏 impl 初始化 prefill backend。
- vllm/model_executor/models/deepseek_v2.py（模块 模型加载；类别 source；类型 data-contract）：GLM/DeepSeek 系列模型权重加载入口，修复了 _try_load_fp8_indexer_wk 对 1-D per-channel FP8 scale 的崩溃，使 Quark 量化 GLM-5.2 checkpoint 可加载。
- vllm/v1/attention/backends/mla/rocm_aiter_mla_sparse.py（模块 稀疏 MLA；类别 source；类型 dependency-wiring）：本次崩溃的直接发生点。补齐了共享 MLA forward 强制读取的 split 字段，并在 build() 中通过 split_decodes_and_prefills 填充，同时声明 supports_dense_mha_prefill = False。
- tests/kernels/attention/test_rocm_aiter_mla_sparse_metadata_sync.py（模块 回归测试；类别 test；类型 test-coverage；符号 _make_mixed_common_metadata, _patch_build_deps, fake_generate_sparse_seqlen_triton, test_build_populates_decode_only_split_fields）：从 #48722 cherry-pick 的回归测试，覆盖 decode-only 与 decode+prefill 混合 batch 下 split 字段的填充语义，是防止该 bug 回退的关键防护。
- vllm/v1/attention/backend.py（模块 后端基类；类别 source；类型 core-logic）：AttentionImplBase 是 MLA 与标准 Attention 的统一基类，本 PR 在此新增 supports_dense_mha_prefill 能力标志，形成所有 impl 的统一能力声明契约。

关键符号：_try_load_fp8_indexer_wk, split_decodes_and_prefills, ROCMAiterMLASparseMetadataBuilder.build, ROCMAiterMLASparseMetadata,

ROCMAiterMLASparseImpl, AttentionImplBase, supports_dense_mha_prefill

关键源码片段

vllm/model_executor/layers/attention/mla_attention.py

共享 MLA forward 的核心文件。#47327 重构后这里新增了从 metadata 读取 split 字段的依赖，本 PR 在 prefill_backend 初始化处引入 supports_dense_mha_prefill 能力分支，从根因上避免无 dense-MHA 的稀疏 impl 初始化 prefill backend。

```
# vllm/model_executor/layers/attention/mla_attention.py
# prefill_backend 初始化：稀疏 MLA impl 若声明不支持 dense-MHA prefill，
# 直接回退到 top-k MQA-only 路径，不再尝试构建 dense-MHA prefill backend。

self.prefill_backend: MLAPrefillBackend | None
if self.impl.is_sparse and not self.impl.supports_dense_mha_prefill:
    logger.warning_once(
        "Sparse MLA impl has no dense-MHA prefill path; using the top-k "
        "MQA path only."
    )
    self.prefill_backend = None
else:
    try:
        prefill_backend_cls = get_mla_prefill_backend(vllm_config)
    except ValueError:
        # 用户显式配置了不支持的 backend 时仍然报错，避免静默降级。
        if (
            not self.impl.is_sparse
            or vllm_config.attention_config.mla_prefill_backend is not None
        ):
            raise
        logger.warning_once(
            "No MLA prefill backend supports this model; sparse MLA will "
            "use the top-k MQA path only (no dense-MHA prefill)."
        )
        self.prefill_backend = None
    else:
        self.prefill_backend = prefill_backend_cls(
            num_heads=self.num_heads,
            scale=self.scale,
            kv_lora_rank=self.kv_lora_rank,
            qk_nope_head_dim=self.qk_nope_head_dim,
            qk_rope_head_dim=self.qk_rope_head_dim,
            v_head_dim=self.v_head_dim,
            vllm_config=vllm_config,
        )
```

vllm/model_executor/models/deepseek_v2.py

GLM/DeepSeek 系列模型权重加载入口，修复了 _try_load_fp8_indexer_wk 对 1-D per-channel FP8 scale 的崩溃，使 Quark 量化 GLM-5.2 checkpoint 可加载。

```
# vllm/model_executor/models/deepseek_v2.py
# 修复点：FP8 indexer wk 的 scale 可能是 1-D per-channel 或 2-D block-wise，
# 原先无条件按 block-wise 推算，遇到 1-D scale 时直接 IndexError。

def _try_load_fp8_indexer_wk(
    name, tensor, buf, params_dict, loaded_params, pp_missing_layer_names
):
    # 仅处理 indexer.wk 的 FP8 权重与 weight_scale_inv，两者都到齐后再反量化。
    if "indexer.wk." not in name or "wk_weights" in name:
        return False
    is_weight = name.endswith(".weight") and tensor.dtype == torch.float8_e4m3fn
    is_scale = "weight_scale" in name
    if not is_weight and not is_scale:
        return False
    # 以 layer 前缀为键缓冲权重与 scale，直到两者都到达。
    layer_prefix = name.rsplit(".wk.", 1)[0]
    fused_name = f"{layer_prefix}.wk_weights_proj.weight"
    if any(name.startswith(m) for m in pp_missing_layer_names):
        return True
    entry = buf.setdefault(layer_prefix, {})
    entry["weight" if is_weight else "scale"] = tensor
    if "weight" not in entry or "scale" not in entry:
        return True # 还差另一个分量，等待下次回调。

    weight_fp8, scale_inv = entry["weight"], entry["scale"]
    del buf[layer_prefix]

    if scale_inv.ndim == 1:
        # 1-D scale：每个输出通道一个 scale，group 覆盖整行（in_features 个元素）。
        group_shape = GroupShape(1, weight_fp8.shape[1])
    else:
        # 2-D block-wise scale：保持原有语义，按列数推算 block 大小。
        block_size = weight_fp8.shape[1] // scale_inv.shape[1]
        group_shape = GroupShape(block_size, block_size)

    weight_bf16 = scaled_dequantize(
        weight_fp8,
        scale_inv,
        group_shape=group_shape,
        out_dtype=torch.bfloat16,
    )
    # 反量化结果写入 fused wk_weights_proj 的 shard 0，保持原有加载契约。
    param = params_dict[fused_name]
    param.weight_loader(param, weight_bf16, 0)
    loaded_params.add(fused_name)
    return True
```

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

本次崩溃的直接发生点。补齐了共享 MLA forward 强制读取的 split 字段，并在 build() 中通过 split_decodes_and_prefills 填充，同时声明 supports_dense_mha_prefill = False。

```
# vllm/v1/attention/backends/mla/rocm_aiter_mla_sparse.py
# dataclass 补齐共享 MLA forward（mla_attention.py）必须读取的 split 字段。
@dataclass
class ROCMAiterMLASparseMetadata(AttentionMetadata):
    num_reqs: int
    max_query_len: int
    max_seq_len: int
    num_actual_tokens: int
    ...
    block_size: int = 1
    topk_tokens: int = 2048

    # 共享 MLA forward 从 metadata 读取这些字段；本 impl 没有 dense-MHA
    # prefill 路径（supports_dense_mha_prefill = False），始终走 MQA 路径，
    # 因此 prefill 相关字段保持默认值即可。
    num_decodes: int = 0
    num_prefills: int = 0
    num_decode_tokens: int = 0
    prefill_max_seq_len: int = 0
    prefill: object = None

    # 持久化 MLA metadata（aiter sparse decode 内核的 work-stealing 分裂）。
    work_meta_data: torch.Tensor | None = None
    ...


# build() 中通过既有辅助函数填充 split 计数。
def build(self, common_prefix_len, common_attn_metadata, fast_build=False):
    num_tokens = common_attn_metadata.num_actual_tokens
    (num_decodes, num_prefills, num_decode_tokens, _) = split_decodes_and_prefills(
        common_attn_metadata,
        decode_threshold=self.reorder_batch_threshold or 1,
    )
    ...
    metadata = ROCMAiterMLASparseMetadata(
        num_reqs=common_attn_metadata.num_reqs,
        max_query_len=common_attn_metadata.max_query_len,
        max_seq_len=common_attn_metadata.max_seq_len,
        num_actual_tokens=common_attn_metadata.num_actual_tokens,
        ...
        num_decodes=num_decodes,
        num_prefills=num_prefills,
        num_decode_tokens=num_decode_tokens,
        ...
```

```
    )
    return metadata
```

评论区精华

- 与 #48722 重复修复的协调：Rohan138 指出已存在另一个修复同一 sparse MLA 问题的 PR（#48722）。ColinZ22 说明 #48722 是本 PR 的子集、本 PR 已 review 通过，最终合并本 PR 并关闭 #48722。
- 测试归属与贡献确认：fanxingran（#48722 作者）请求将专属回归测试 cherry-pick 进本 PR，并保留 co-author trailer；ColinZ22 接受并合入该测试。
- 流程提醒：tjtanaa 与 Rohan138 多次提醒不要反复 merge main（CI 已通过，merge 只引入噪音和额外失败风险）。
- 独立确认与后续工作：jimmy-adams 在 MI350 上独立复现同一 crash 并确认修复有效，同时报告 NVFP4 MTP/nextn draft 层 bf16 加载崩溃（#49275），与本 PR 正交；tjtanaa 批准时明确"接受热补丁，后续跟进为 rocm aiter sparse mla 实现 dense-MHA 路径"。
  - 与 #48722 重复修复的合并协调 (design): 合并 #48886 并关闭 #48722，fanxingran 的回归测试 cherry-pick 进本 PR。
  - 回归测试归属与 cherry-pick (testing): ColinZ22 接受并合入测试，commit 中保留 Co-authored-by。
  - 避免反复 merge main (other): 作者停止 merge，由 reviewer 重试失败 job 后合并。
  - ROCm sparse MLA 的 dense-MHA prefill follow-up (design): 接受当前 MQA-only 回退方案，后续单独实现 forward_mha 并同步填充 prefill_max_seq_len 等字段。
  - NVFP4 MTP/nextn draft 层 bf16 加载崩溃（#49275） (correctness): 作为正交问题移交 #49275 处理，与本 PR 互补。

风险与影响

- 风险：
  - 能力标志默认值风险：supports_dense_mha_prefill 默认 True，若未来新增稀疏 MLA impl 忘记覆盖该标志且未实现 forward_mha，会再次落入 dense-MHA prefill 路径并崩溃；这是一个"显式声明能力"的约定，需要 review 时把关。
  - per-channel scale 语义假设：deepseek_v2.py 将 1-D scale 一律解释为 per-output-channel（GroupShape(1, in_features)），若未来出现其他 1-D 语义（如 per-token）的 checkpoint，会静默反量化错误；该分支目前没有专门的加载路径单测。
  - 字段默认值与后续 dense-MHA 路径：MQA-only 语义下 prefill_max_seq_len == 0、prefill is None 是合理的；tjtanaa 计划中的 dense-MHA prefill follow-up 若只填充部分字段，会破坏共享 forward 的读取契约。
  - 测试平台门控：新回归测试依赖 aiter 环境（ROCm 门控），常规 CI 不覆盖；deepseek_v2.py 的 per-channel 修复也缺少防护测试。
- 影响：
  - 用户侧：ROCm 上 Quark 量化 GLM-5.2（Attn FP8）checkpoint 从完全不可运行变为端到端可用，gsm8k（5-shot）准确率 0.9295；同类 MLA 稀疏模型（如

GLM-5.1-NVFP4 + MTP）同样受益。

- 系统侧：v1 MLA attention 的 metadata 契约在 ROCMAiterMLASparseMetadata 上补齐，与 FlashMLA / FlashAttn sparse 后端对齐；supports_dense_mha_prefill 成为所有 MLA impl 家族的统一能力声明点，后续新增后端需显式声明。
- 团队侧：与 #48722 合并避免了两个重复修复并存，回归测试来自社区贡献者并保留 co-author，属于多贡献者协作的典型案例。
- 风险标记：核心路径变更，能力标志默认值依赖约定，per-channel 分支缺少加载单测，ROCm 门控测试，MQA-only 语义对 prefill 字段的默认值依赖

关联脉络

- PR #48722 稀疏 MLA 元数据修复的重复 PR（讨论提及）：fanxingran 提交的重复修复 PR，与本 PR 覆盖同一 ROCmAiterMLASparseMetadata 缺失字段问题；其回归测试被 cherry-pick 进本 PR 并保留 co-author。
- PR #47327 共享 MLA forward 重构（讨论提及）：重构了 mla_attention.py 的共享 forward，使其从 attention metadata 读取 num_decodes、num_prefills、num_decode_tokens、prefill_max_seq_len；本 PR 修复的正是该重构未同步 ROCm aiter sparse 后端导致的崩溃。
- PR #49275 NVFP4 MTP/nextn draft 层 bf16 加载崩溃（讨论提及）：jimmy-adams 报告的 Quark 加载层正交问题：NVFP4 checkpoint 的 MTP 草稿层以 bf16 存储且缺 exclude 条目，与本 PR 修复的 indexer wk 路径相互独立但同属 GLM 系列量化 checkpoint 加载链路。