

# PR #51040 完整报告

vllm-project/vllm

[ROCm][K3] Extend FP8 asm MLA prefill to non-divisor small head counts

合并时间: 2026-08-27 01:47

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51040>

## 执行摘要

- 一句话: K3 FP8 MLA prefill 支持非 16 整除小头数, 长上下文不再 OOM
- 推荐动作: 值得精读。这是 ROCm MLA FP8 推理栈中「kernel 对齐要求」与「真实模型头数」矛盾的典型案例, 展示了如何用 replicate-pad/slice-back 以少量拷贝换取 asm kernel 的启用, 以及如何通过统一 helper 避免 metadata 与 forward 宽度漂移。关注 `get_fp8_prefill_num_heads` 的设计、workspace 化输出 scratch 的做法, 以及 review 中关于 `num_real_heads` in-kernel 支持的后续展望。

## 功能与动机

Kimi-K3 在 TP8 下每 rank 仅有 12 个 attention heads, 而 AITER FP8 MLA asm prefill (`mha_prefill_ps_asm_fwd + mha_reduce_v1`) 原门槛要求 `num_heads % 16 == 0`, 导致 K3 回退到 BF16 FMHA 解压路径; 该回退构建的 bf16 工作集不在 FP8 KV 池 accounting 范围内, 长上下文 (约 197k tokens) 时激活 arena 耗尽 OOM, 而 KV pool 使用率不足 4%。PR body 明确指出这是 #50578 (asm decode pad 12→16) 的 prefill 对应部分, 利用 MLA 每个 query head 在共享 latent KV 上独立计算的性质, 将 head 填充到 16 再切回是精确操作。

## 实现拆解

1. 放宽 `_fp8_prefill_enabled` 门槛 (`vllm/v1/attention/backends/mha/rocm_aiter_mha.py`)  
: 由 `_fp8_mha_prefill_supported()` and `self.num_heads % 16 == 0` 改为同时要求 `kv_cache_dtype_str == "fp8"`、`model_config.dtype == torch.bfloat16` 和 `AiterMLAHelper.is_valid_num_heads(self.num_heads)`; 这样 bf16 KV 服务不会预留 PS workspace, fp16 模型也不会把 bf16 输出位模式误读为 fp16 (review 中发现的真实 gap)。
2. 统一 head pad 目标到 helper: 新增 `AiterMLAHelper.get_fp8_prefill_num_heads`, 并让 `_init_fp8_prefill_ps_buffers` 与 `_build_fp8_prefill_ps_metadata` 的 `num_head_k` 从 `max(16, num_heads)` 改为该 helper 结果, 使 work/reduce maps 与 forward 实际 pad 宽度一致; 该改动同时降低 partial tiles (`gcd` 从 4 变 16), 实测 `batch=1/qlen=512` 时保留内存从 193.6 MiB 降到 68.6 MiB, 为长上下文节省约 6 GiB。
3. `_mha_fp8_prefill_attn` 内 `replicate-pad` 与 `slice-back`: 对非 16 倍数小头数 (如 12) 用 #50578 引入的 `get_mha_padded_q` 将 Q/K/V pad 到 16, 运行 PS asm prefill 与 `mha_reduce_v1` 后把输出切回真实 `num_heads`。

4. workspace 化输出 scratch: out\_3d 从每次 torch.empty 改为从全局 workspace manager 的 get\_simultaneous 调用中获取, 与 logits/attn\_lse/final\_lse 同来源, 避免 per-call 动态分配, 并在 \_init\_fp8\_prefill\_ps\_buffers 中预留最大形状; 同时用 ATTN\_OUT\_DTYPE = torch.bfloat16 常量统一输出缓冲与 workspace 预留 dtype。
5. 测试覆盖: tests/kernels/attention/test\_rocm\_aiter\_mla\_fp8\_prefill.py 将单一 NUM\_HEADS = 16 改为 HEAD\_COUNTS = [12, 16, 24], 覆盖“非 divisor pad 到 16”“无需 pad 输出别名”“pad 到 32”三种 regime; \_make\_impl 与 \_build\_prefill\_metadata 增加 num\_heads 与 attn\_out\_dtype 参数, 新增 test\_fp8\_prefill\_matches\_reference 参数化与 test\_fp8\_prefill\_metadata\_width\_matches\_forward 元数据宽度一致性用例。

关键文件:

- vllm/v1/attention/backends/mla/rocm\_aiter\_mla.py (模块 注意力后端; 类别 source; 类型 core-logic; 符号 get\_mla\_padded\_q, get\_fp8\_prefill\_num\_heads, \_mla\_fp8\_prefill\_attn, \_init\_fp8\_prefill\_ps\_buffers): 核心实现文件: 放宽 FP8 prefill 头数门槛、新增统一 pad 头数 helper、将 PS metadata/workspace 预留与 forward pad 目标对齐、用 workspace manager 承载 padded-head 输出 scratch。
- tests/kernels/attention/test\_rocm\_aiter\_mla\_fp8\_prefill.py (模块 kernel 测试; 类别 test; 类型 test-coverage; 符号 \_make\_impl, \_build\_prefill\_metadata, test\_fp8\_prefill\_matches\_reference, test\_fp8\_prefill\_metadata\_width\_matches\_forward): 测试覆盖从单一 16 头扩展到 12/16/24 三种 regime, 验证 pad/slice 正确性与 metadata 宽度与 forward 一致, 并顺带暴露 fp16 模型误用 bf16 输出的缺陷。

关键符号: get\_fp8\_prefill\_num\_heads, \_mla\_fp8\_prefill\_attn, \_init\_fp8\_prefill\_ps\_buffers, \_build\_fp8\_prefill\_ps\_metadata, get\_mla\_padded\_q

## 关键源码片段

### vllm/v1/attention/backends/mla/rocm\_aiter\_mla.py

核心实现文件: 放宽 FP8 prefill 头数门槛、新增统一 pad 头数 helper、将 PS metadata/workspace 预留与 forward pad 目标对齐、用 workspace manager 承载 padded-head 输出 scratch。

```
# vllm/v1/attention/backends/mla/rocm_aiter_mla.py (整理后核心片段)
```

```
def get_fp8_prefill_num_heads(num_heads: int) -> int:
```

```
    """FP8 asm prefill 与 mla_reduce_v1 要求 16 对齐的头数。
```

```
    对非 divisor 的小头数 (如 K3 在 TP8 下每 rank 12 头),
    _mla_fp8_prefill_attn 会把 Q/K/V replicate-pad 到 16 头再 slice 回来;
    PS metadata 的 num_head_k 也必须用同一个目标宽度, 否则 work/reduce
    maps 与 kernel 实际拿到的张量宽度不一致。统一走这里, 同时避免
    max(16, num_heads) 在 24 头时产生更碎的 partial tiles。
    """
```

```
    return max(_AITER_MIN_MLA_HEADS, num_heads) # 16 的倍数向上取整
```

```
# __init__ 中的启用门槛: FP8 KV + bf16 模型 + 有效头数, 缺一不可
```

```

self._fp8_prefill_enabled = _fp8_mla_prefill_supported() and (
    kv_cache_dtype_str == "fp8"
    and vllm_config.model_config.dtype == torch.bfloat16
    and AiterMLAHelper.is_valid_num_heads(self.num_heads)
)

```

```

# _init_fp8_prefill_ps_buffers 中: metadata 按 pad 后的宽度构造
# 之前是 num_head_k = self.num_heads; 对 24 头会保留 24 宽 work maps,
# 但 forward 实际 pad 到 32, 导致 64 个 partial tiles (约 193.6 MiB) ;
# 统一用 helper 后取 32, partial tiles 降到 16 (约 68.6 MiB, gfx950 实测)。
num_head_k = AiterMLAHelper.get_fp8_prefill_num_heads(self.num_heads)

```

## tests/kernels/attention/test\_rocm\_aiter\_mla\_fp8\_prefill.py

测试覆盖从单一 16 头扩展到 12/16/24 三种 regime, 验证 pad/slice 正确性与 metadata 宽度与 forward 一致, 并顺带暴露 fp16 模型误用 bf16 输出的缺陷。

```

# tests/kernels/attention/test_rocm_aiter_mla_fp8_prefill.py (整理后核心片段)

# 覆盖三种 head 数 regime:
# 12 -> pad 到 16 (非 divisor, tile+slice) ——K3 在 TP8 下的真实场景
# 16 -> 无需 pad, 输出直接别名调用方 buffer
# 24 -> pad 到 32, 证明 16 以上也存在 pad 路径 (当前无模型触发, 防未来回归)
HEAD_COUNTS = [12, 16, 24]

# asm kernel 以原始指针写 bf16, 输出缓冲与 workspace 预留必须同 dtype
ATTN_OUT_DTYPE = torch.bfloat16

@pytest.mark.parametrize("num_heads", HEAD_COUNTS)
@pytest.mark.parametrize("seq_len", [128, 512])
@torch.inference_mode()
def test_fp8_prefill_matches_reference(num_heads: int, seq_len: int) -> None:
    # 用生产 builder 构建 metadata, 避免手搓导致与实现漂移
    metadata, total_q, builder = _build_prefill_metadata(
        [seq_len], device, num_heads, ATTN_OUT_DTYPE
    )
    # ... 构造 q/k/v, 调用 _mla_fp8_prefill_attn, 与 fp8 量化输入的
    # causal SDPA 参考实现对比, atol=1e-1 / rtol=5e-2 覆盖 fp8 量化噪声

def test_fp8_prefill_metadata_width_matches_forward(num_heads: int) -> None:
    # 断言 builder 中 fp8_prefill_* 的宽度 == get_fp8_prefill_num_heads(num_heads),
    # 防止 metadata sizing 与 forward pad 目标脱钩 (此前 max(16, n) 在 24 头时失配)
    ...

```

## 评论区精华

- 输出 copy 能否消除 (Rohan138) : 询问能否后续让 AITER 在 kernel 内支持 padding+slicing, 并关注 out\_3d 初始化对 CUDAGraph/ 持久 MLA/prefix caching 的影响。作者回应: out 是按 12 头打包的, kernel 写 16 头行, 任何 view 都无法直接接收结果, 必须 scratch+slice-back; 输出 copy 仅约 6 KB/token, 而 Q/K/V padding 约 28

KB/token, num\_real\_heads 参数可一次性消除四份拷贝；同时 out\_3d 已改为从 workspace manager 获取。

- 测试中硬编码 dtype (AndreasKaratzas)：质疑测试里为何硬编码 `attn_out_dtype=torch.bfloat16`。作者承认并修复，还顺带发现非测试路径的真实缺陷：fp16 模型 + fp8 KV cache 会走该路径导致 bf16 位被当作 fp16 解释，于是新增 `model_config.dtype == torch.bfloat16` 门禁。
- TP<8 (如 TP4 时 24 头) 是否启用 (shen-shanshan)：作者调研后表示 K3 因权重容量无法跑 TP4，并逐一核对现有 MLA 模型 (K2 64、LongCat-Flash 64、Kimi-VL 16、DeepSeek-V3/R1 128) 在各 TP 下均不会落入 16 以上非倍数区间；MiniMax-M2 (48) 与 GLM-4.6 (96) 虽会落入但均为 GQA 而非 MLA，因此当前无实际触发场景，但泛化改动仍值得做。
- 硬编码 16 的健壮性 (shen-shanshan)：建议把所有 16 换成 `AiterMLAHelper._AITER_MIN_MLA_HEADS`。作者接受并进一步发现 PS metadata sizing 中的 `max(16, num_heads)` 也需要与 forward pad 目标联动，因此统一收敛到 `get_fp8_prefill_num_heads` 一个 helper。
- padded-head 输出 copy 能否消除 (design)：作者回应 copy 无法从 vLLM 侧消除 (out 是 12 头打包, kernel 写 16 头行)，但输出 copy 仅约 6 KB/token, Q/K/V padding 约 28 KB/token；AITER 加 num\_real\_heads 参数可一次消除四份拷贝，可开 follow-up。out\_3d 已改为从 workspace manager 获取。
- 测试中硬编码 bf16 与 fp16 模型误用 (correctness)：测试使用 `ATTN_OUT_DTYPE` 常量；`_fp8_prefill_enabled` 新增 `model_config.dtype == torch.bfloat16` 门禁，fp16 模型回退到 BF16 prefill 路径。
- TP<8 时 24 头是否启用 asm prefill (question)：作者核实 K3 无法跑 TP4 (权重容量)，并逐一检查现有 MLA 模型 (K2 64、LongCat-Flash 64、Kimi-VL 16、DeepSeek-V3/R1 128) 在各 TP 下都不会落入 16 以上非倍数 band；MiniMax-M2/GLM-4.6 虽会但为 GQA。泛化改动仍值得。
- 硬编码 16 的健壮性 (design)：作者接受并把 PS metadata sizing 的 `max(16, num_heads)` 一并收敛到 `get_fp8_prefill_num_heads`，避免 forward pad 目标与 metadata 宽度脱钩。

## 风险与影响

- 风险：
  - fp16 模型误用 bf16 输出：asm prefill 以原始指针写 bf16，若 fp16 模型启用该路径会把 bf16 位模式重解释为 fp16 产生静默错误；已通过 `model_config.dtype == torch.bfloat16` 门禁修复，但依赖该门禁在后续重构中不被移除。
  - pad/slice 的数值与形状一致性：replicate-pad 依赖 MLA 每 head 独立计算的语义，若未来引入跨 head 的 kernel 或 MLA 结构变化，pad 后 slice 的精确性可能被破坏；测试覆盖 12/16/24 三种头数，但 24 头场景当前无真实模型。
  - workspace 预留变化：num\_head\_k 从 `max(16, num_heads)` 改为 `get_fp8_prefill_num_heads` 后，workspace 预留显著变小 (约 6 GiB)，若 PS metadata 与 forward 的 pad 目标不同步，会出现越界或预留不足；测试

test\_fp8\_prefill\_metadata\_width\_matches\_forward 正是针对此风险。

- CUDAGraph/prefix caching 交互：out\_3d 改为从 workspace manager 获取后，其地址稳定性依赖 workspace manager 的持续性；review 中 Rohan138 对此提出过疑虑，作者称已与 logits 等等来源，但 CUDAGraph 捕获场景仍需回归验证。
- CUDA graph 与 breakable cudagraph：PR body 的 serve 命令显式设置 VLLM\_USE\_BREAKABLE\_CUDAGRAPH=0，说明该路径与 breakable cudagraph 的兼容性尚未完全验证，部署配置受限。
- 影响：对 Kimi-K3 on ROCm gfx950 (MI355X) 用户是实质性修复：长上下文 prefill 从 OOM (约 197k tokens) 提升到 470k/590k tokens 可正常完成，PS workspace 节省约 6 GiB，GSM8K 5-shot 准确率 97.04% 与基线 96.66% 相比无回退（在噪声范围内）。对其他 MLA 模型（K2、LongCat-Flash、Kimi-VL、DeepSeek-V3/R1）当前各 TP 下头数均为 16 的倍数或 16 以下，行为不变但门槛放宽后未来非 divisor 头数模型也可自动受益。对团队而言，该 PR 确立了“头数 pad 目标与 PS metadata sizing 必须共用一个 helper”的设计约束，并沉淀了 kernel 级数值对比测试与元数据宽度测试，降低后续回归风险。测试文件仅 gfx950 + AITER 构建可用，其他平台自动 skip。
- 风险标记：核心路径变更，平台特定（gfx950/AITER），新增门禁依赖，workspace 预留敏感，CUDAGraph 交互待验证

## 关联脉络

- PR #50578 [ROCm][MLA] Support non-divisor small head counts in asm decode (pad 12 to 16): 本 PR 明确是 #50578 的 prefill 对应部分，复用了其 AiterMLAHelper.get\_mla\_padded\_q 与头数 pad 语义；依赖其先合并。
- PR #48712 [ROCm][MLA] Gate PS workspace on FP8 KV cache: 本 PR 在 \_fp8\_prefill\_enabled 中加入了与 #48712 相同的 FP8 KV 门禁（避免 bf16 KV 服务预留 PS workspace），PR body 说明与该 PR 组合使用。