

PR #51171 完整报告

vllm-project/vllm

[ROCm][MLA] Reach FULL cudagraphs for AITER MLA speculative decoding

合并时间: 2026-08-30 10:49

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51171>

执行摘要

- 一句话: ROCm MLA 投机解码实现 FULL cudagraph, 吞吐提升约 3.8 倍
- 推荐动作: 值得精读。这是 ROCm MLA 投机解码性能的关键一跃, 且 review 质量高: okorzh-amd 的拆分并行度分析、tjtanaa 对 4-D 入口的提示都直接改变了最终方案。建议重点学习四个设计决策: builder 与 impl 的职责边界 (同步必须离开捕获区)、组级而非层级的 CG 支持上报、用 kernel 原生 4-D MTP 替代手动 flatten、以及在修复性能时同步修正正确性隐患 (min_kv_seq_len 污染、assert 剥离、padding 行溢出)。

功能与动机

PR body 明确指出: "Speculative decoding on ROCm cannot reach FULL cudagraphs, for two independent reasons. This fixes both." 第一个原因是类常量 CG 支持上报: 一个服务 $1 + \text{num_speculative_tokens}$ 块的 DSpark draft 组让引擎取最小值后降级整个引擎; 第二个原因是 AiterMLAIml.forward_mqa 小 head verify 路径每层做 4 次 device->host 同步和数据相关分配, 每个都会中止 HIP graph capture。此外还发现 worker 侧一直缺失 kv_cache_size_tokens, 导致 verify buffer 按 max_model_len 过度预留 (295.3 MiB vs 修正后 46.6 MiB)。reger-men 在 gfx950 上独立复现同样的两个缺陷, 并给出 spec-off 时强制 PIECEWISE 也要付出 5.2 倍代价的数据, 证明 decode graph 本身价值极高。

实现拆解

整个变更分四条线索推进, 共 4 个文件、12 个 commit, 期间经历了从 "手动展开" 到 "kernel 原生 4-D 入口" 的方案升级:

1. 修复 CG 支持上报 (文件: vllm/v1/attention/backends/mla/triton_mla.py)。为 TritonMLAMetadataBuilder 新增类方法 get_cudagraph_support, 当 kv_cache_spec 带有 non_causal_multi_token_decode 标记时返回 UNIFORM_BATCH, 否则回退原类常量。该标记是 KV cache 组级属性 (MLAAttentionSpec.merge 会对组内所有层做 OR), 与 __init__ 里提升 reorder_batch_threshold 用的是同一谓词, 保证两处判定一致; 副作用是同组的 causal target 也会被一并提升, 已在 docstring 中记录作用域。
2. 重构小 head verify 路径 (文件: vllm/v1/attention/backends/mla/rocm_aiter_mla.py)。早期 commit(242c0974) 先把每层展开移到 _build_decode 一次完成并写入持久 buffer; 随后 tjtanaa review 指出 aiter 已支持 4-D 张量, 作者在 f3c0eb56 改为走 mla_gluon 原生 4-D MTP 入口 :q/o unflatten 成 [num_reqs, qlen, nhead, dim], 直接传每请求的 paged_kv_indptr 与 paged_kv_indices, 因果上界由 aiter v0.1.20 在 kernel 内施加, 彻

底删除 flat_kv_indptr/flat_kv_indices 持久 buffer、_expand_page_indices_kernel 的 QLEN 展开逻辑和 builder 侧对应同步。

3. 修正 min_kv_seq_len 语义并对边界加固。okorzh-amd 指出 FULL-cudagraph 的 pad_uniform_mtp 会把零 qo 行钉到 max_qo_len, 导致 per_req_len.min() 返回 qlen 而非真实 KV 长度, 把 Gluon KV-split 并行度从 17 路压到 1 路; 修复为仅对 qo_lens_device > 0 的 active 行取 min, 并新增 test_min_kv_seq_len_ignores_cudagraph_padding_rows。同时把 flat buffer 的 assert 边界检查改为 raise ValueError(python -O 会剥离 assert, 越界时 Triton kernel 会写坏相邻显存)。
4. 配套测试与清理。test_rocm_aiter_mla_causal_verify_mask.py 从 " 逐行因果窗口长度 " 断言改为钉 4-D MTP 契约 (q 4-D、use_2d_view=False、1-D page_table、[num_reqs+1] seq_info); test_rocm_aiter_mla_mtp_split.py 的 _ExpandPageIndicesKernel mock 同步改为从 indptr 差分推导行数。gpu_worker.py 的 kv_cache_size_tokens 填充曾临时加入 ,4-D 路径不再依赖后由 shen-shanshan 指出并移除。

关键文件:

- vllm/v1/attention/backends/mla/rocm_aiter_mla.py (模块 MLA 后端; 类别 source; 类型 core-logic; 符号 _build_decode, forward_mqa, use_gluon_verify, _expand_page_indices_kernel): 核心变更文件: 小 head verify 从每层 flat 展开改为 mla_gluon 原生 4-D MTP 入口, min_kv_seq_len 计算移入 builder 并排除 padding 行, 边界检查从 assert 改为 raise ValueError。
- vllm/v1/attention/backends/mla/triton_mla.py (模块 图捕获; 类别 source; 类型 dependency-wiring; 符号 get_cudagraph_support): 新增 get_cudagraph_support 类方法, 按 KV cache 组属性上报 UNIFORM_BATCH, 消除整个引擎因单个 draft 组被降级出 FULL cudagraph 的问题。
- tests/kernels/attention/test_rocm_aiter_mla_causal_verify_mask.py (模块 回归测试; 类别 test; 类型 test-coverage; 符号 test_verify_mtp_uses_native_4d_gluon_entry, _run_verify_block): 回归测试从 " 逐行因果窗口长度 " 改为钉 4-D MTP 契约, 防止重蹈 verify 位置能看到其后的 draft token 的正确性缺陷。
- tests/v1/attention/test_rocm_aiter_mla_mtp_split.py (模块 ROCm 测试; 类别 test; 类型 test-coverage; 符号 test_min_kv_seq_len_ignores_cudagraph_padding_rows, _ExpandPageIndicesKernel): 新增 test_min_kv_seq_len_ignores_cudagraph_padding_rows, 防止 FULL 图捕获下 padding 行把 Gluon KV-split 并行度压垮; 同步更新 kernel mock 签名。

关键符号: TritonMLAMetadataBuilder.get_cudagraph_support,
AiterMLAMetadataBuilder._build_decode, AiterMLAImpl.forward_mqa,
AiterMLAHelper.use_gluon_verify, _expand_page_indices_kernel,
test_verify_mtp_uses_native_4d_gluon_entry,
test_min_kv_seq_len_ignores_cudagraph_padding_rows

关键源码片段

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

核心变更文件：小 head verify 从每层 flat 展开改为 mla_gluon 原生 4-D MTP 入口，min_kv_seq_len 计算移入 builder 并排除 padding 行，边界检查从 assert 改为 raise ValueError。

```
# 小 head (< 16) 多 token verify 走 mla_gluon 的 4-D MTP 入口，直接使用
# 每请求的 paged-KV 视图，不再构建逐 verify token 的展开 buffer。
# mla_gluon 仍需要一个 KV 长度下界用于决定内部分割；verify 路径上该下界
# 必须基于真实请求，而不能被 FULL cudagraph 的占位行污染。
# (pad_uniform_mtp 会把零 qo 行钉到 max_qo_len，若对全部行取 min 会得到
# qlen 而非真实 KV 长度，进而把 Gluon 的 KV-split 并行度从 17 路压到 1 路。)
min_kv_seq_len = 1
if AiterMLAHelper.use_gluon_verify(
    self.num_heads, int(max_qo_len), self._kv_cache_dtype_str
):
    # paged_kv_indptr 的差分即每请求 KV 行数
    per_req_len = paged_kv_indptr[1:] - paged_kv_indptr[:-1]
    if pad_uniform_mtp:
        # 只统计真实请求 (qo_len > 0)，忽略 cudagraph 占位行
        active = qo_lens_device > 0
        if active.any():
            # .item() 在 builder 内执行，位于捕获区之外，不会中止 HIP 图捕获
            min_kv_seq_len = int(per_req_len[active].min().item())
    else:
        min_kv_seq_len = int(per_req_len.min().item())

if AiterMLAHelper.use_gluon_verify(
    self.num_heads, int(decode.max_qo_len), self.kv_cache_dtype
):
    # 小 head (< 16) 多 token verify 交给 mla_gluon 的原生 4-D MTP 入口：
    # q 从 [B, nhead, dim] 还原成 [num_reqs, qlen, nhead, dim](flat 布局
    # 是 row-major, unflatten 是零成本 view)，因果上界由 aiter v0.1.20
    # 在 kernel 内施加 :score_end = min(split_kv_end, seq_len - qlen + q_pos + 1)。
    # 相比旧实现，这里不再按 verify token 展开 paged-KV，也没有每层
    # 4 个 device -> host 同步和数据相关分配，因此可被 HIP 图捕获。
    qlen = int(decode.max_qo_len)
    num_reqs = B // qlen
    if num_reqs * qlen != B:
        raise ValueError(f"verify block {B} rows is not a multiple of qlen {qlen}")
    assert decode.paged_kv_indptr is not None
    assert decode.paged_kv_indices is not None
    mla_gluon = _get_mla_gluon()
    mla_gluon(
        q_nope=q_nope.unflatten(0, (num_reqs, qlen)),
        q_pe=q_pe.unflatten(0, (num_reqs, qlen)),
        kv_c=kv_buffer,
        o=o.unflatten(0, (num_reqs, qlen)),
        page_table=decode.paged_kv_indices,
        seq_info=decode.paged_kv_indptr,
        sm_scale=self.scale,
```

```

        k_pe=None,
        kv_pe_offset=self.kv_lora_rank,
        use_2d_view=False,
        kv_scale=1.0,
        min_kv_seq_len=decode.min_kv_seq_len,
    )
    return o, None

```

tests/kernels/attention/test_rocm_aiter_mla_causal_verify_mask.py

回归测试从 " 逐行因果窗口长度 " 改为钉 4-D MTP 契约，防止重蹈 verify 位置能看到其后的 draft token 的正确性缺陷。

```

def test_verify_mtp_uses_native_4d_gluon_entry():
    """验证 4-D MTP 契约:verify 必须以 4-D q + per-request 元数据进入 Gluon。

```

回归保护:旧实现给每个 flatten 行发整段请求 paged-KV 范围,verify 位置 t 能看到位于其后的 draft token(即它本应校验的 token)。原生 MTP 路径把因果掩码委托给 aiter kernel。

```

    """

```

```

    metadata, captured = _run_verify_block()
    decode = metadata.decode
    assert decode is not None, "batch was not classified as a decode"
    assert decode.max_qo_len == QLEN

```

```

    q_nope = captured["q_nope"]
    assert q_nope.dim() == 4, f"verify 必须以 4-D q 调用 mla_gluon"
    assert q_nope.shape[:2] == (num_reqs, QLEN)
    assert captured["use_2d_view"] is False
    # per-request 元数据 :1-D page_table + [num_reqs + 1] indptr,
    # 而不是按 verify token 展开的扁平视图
    assert captured["page_table"].dim() == 1
    assert captured["seq_info"].numel() == num_reqs + 1

```

评论区精华

评论区最有价值的交锋集中在三处：

- min_kv_seq_len 被 padding 行污染 (okorzh-amd, 正确性 / 性能): "min_kv_seq_len is taken over all requests including cudagraph padding rows. Under pad_uniform_mtp those rows are pinned to max_qo_len... so the min is qlen, not a real KV length." 他给出拆解 :7 个 padding 行把 NUM_KV_SPLITS 从 17 压到 1, 并指出 PR 测出的性能收益是在 split 并行被关掉的情况下测的。作者确认修复于 afd132cfd, 重测 c1 为 45.33 tok/s, 说明修复没有损害收益。
- 放弃手动展开、改用原生 4-D MTP 入口 (tjtanaa, 设计): "Do we still need to prepare this flat_kv_indptr? The aiter v0.1.19 documents that it accepts 4D tensors now." 这一问直接促成了方案从 " 每层 / 每步展开 paged-KV " 重构成 "kernel 内因果掩码 " 的简化, 删除了大量持久 buffer 与同步代码。

- assert 在 python -O 下被剥离 (depthfirst-app + claude[bot], 安全): "If stripped, the subsequent Triton kernel _expand_page_indices_kernel writes to flat_kv_indices... corrupting adjacent GPU memory." 作者改为与 kv_cache_spec_registry.py 一致的 raise ValueError 模式。claude[bot] 还指出 flat buffer 预留未计入 padding 行可能溢出, 该风险随 flat 路径删除而消除。

另有两条低强度评论 :shen-shanshan 纠正 docstring 中 use_2d_view=True 应为 False, 并质疑 gpu_worker.py 的 kv_cache_size_tokens 填充是否还有用; 作者确认是旧 flat 路径遗留, 已在 0abdc05c 移除。

- min_kv_seq_len 被 cudagraph padding 行污染导致 KV-split 并行塌缩 (correctness): 作者采纳, 改为 pad_uniform_mtp 时仅对 qo_lens_device > 0 的 active 行取 min, 新增单测; 重测 c1 为 45.33 tok/s, 收益未受损。
- 是否改用 aiter 原生 4-D MTP 入口替代手动 flat 展开 (design): 作者在 f3c0eb56 采纳: verify 直接传 per-request paged_kv_indptr/paged_kv_indices 与 4-D q, aiter v0.1.20 在 kernel 内施加因果掩码, flat buffer 及相关代码全部删除。
- assert 边界检查在 python -O 下被剥离导致越界写 (security): 作者改为显式 if/raise ValueError; 后续 4-D 路径删除 flat buffer, 该风险整体消除。
- flat_kv_indices 预留容量未计入 padding 行可能溢出 (correctness): 该 flat 展开路径随后被 4-D MTP 入口整体替换, 缓冲区不复存在, 风险随代码删除而消除。
- worker 侧 kv_cache_size_tokens 填充是否还需要 (question): 作者确认是旧 flat 路径遗留, 4-D 路径不再读取 worker 上的 kv_cache_size_tokens, 已在 0abdc05c 移除填充逻辑。
- use_gluon_verify docstring 中 use_2d_view 方向错误 (documentation): 作者在 0abdc05c 连同测试模块 docstring 一起修正为 use_2d_view=False。

风险与影响

- 风险:
 - 与 AITER 版本行为强耦合: 4-D MTP 路径把因果掩码委托给 aiter v0.1.20 的 kernel, min_kv_seq_len 的语义也随 ROCm/aiter#4555 变化 (bh16 不再用它决定拆分, 仅 bh64 作下界断言)。若用户环境安装的 aiter 版本低于 v0.1.20, 性能与正确性依赖将不一致, 建议在文档或环境校验中明确最低版本。
 - 组级 CG 上报的连带提升: get_cudagraph_support 按 KV cache 组返回 UNIFORM_BATCH, 与 draft 共享组的 causal target 也会被提升。虽然与 reorder_batch_threshold 共用同一谓词、且 forward_mqa 无同步保证, 但该分支在真实 causal target 上的覆盖仍主要依赖单元测试, 建议关注线上日志中 FULL 模式下 target 端的行为。
 - FULL 捕获内存显著上升: capture memory per rank 从 0.51 GiB 升到 2.11 GiB, 低显存配置下可能 OOM; 这是 FULL cudagraph 的固有成本, 但部署文档应提示。
 - fp8 KV 显式排除: use_gluon_verify 对量化 KV 返回 False, 避免触发 Gluon fp8 regime 的 batch_size==1 断言; fp8 用户不会获得本 PR 的 verify 加速, 需要明确预期。
 - worker 侧 kv_cache_size_tokens 缺口仍在: 该字段在 worker 上仍为 None 的既有问题只是不再影响 verify buffer (4-D 路径不依赖), 其他据此做容量估算的代码仍可能受影响。

- 影响:

- 用户影响 (高): ROCm + AITER MLA + DSpark 类投机解码 (Kimi-K3、DeepSeek 系列在 TP8/TP16 下小 head 场景) 首次获得 FULL cudagraph。c1 吞吐 12.10 -> 46.30 tok/s, ITL 186.98 -> 34.26 ms; c16 吞吐 174.01 -> 326.46 tok/s。reger-men 在 gfx950 上独立复现: spec on 从 15.12 到 89.22 tok/s, 且 spec-off 时 PIECEWISE 对比 FULL 也要付出 5.2 倍代价。GSM8K 精度 0.9651 -> 0.9629(误差带内), 接受率基本不变, 说明收益全部来自执行速度。
- 系统影响 (中): FULL 图捕获需要额外约 1.6 GiB/rank 显存; 图模式从 PIECEWISE 升至 FULL, 运行期行为与日志 (warning 消失) 发生变化。
- 团队影响 (中): 该 PR 确立了三条可复用的后端设计模式: 把捕获区内的同步移到 builder、用组级 KV 属性驱动 CG 支持上报、优先使用 kernel 原生批量入口而非手动展开。对后续 ROCm 其他 attention backend 走向 FULL cudagraph 有直接参考价值。
- 风险标记: 图捕获核心路径, 依赖 AITER v0.1.20 行为, 捕获内存增长明显, 组级 CG 上报范围

关联脉络

- PR #50578 [ROCm][MLA] (同文件关联 PR, 标题以仓库实际为准): PR body 明确说明 " #50578 touches the same file and is best reviewed alongside. They merge cleanly in either order and do not collide on the verify path.", 与本次改动共享 rocm_aiter_mla.py 的 verify 路径, 需并行评审确认无行为冲突。