

PR #55966 完整报告

vllm-project/vllm

[ROCm][Spec Decode] Add Aiter MLA decode support non-causal draft block

合并时间: 2026-09-16 01:18

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/55966>

执行摘要

- 一句话: ROCm Aiter MLA 支持非因果 draft 解码
- 推荐动作: 值得精读, 尤其是维护 ROCm/MLA 后端的同学。建议按三条线索阅读: (1) `AiterMLABackend.supports_non_causal()` 的「Python 签名 + per-arch manifest」双层判定——它说明了 AITER 的 Python 接口与 ASM kernel 表不同步这一结构性问题; (2) `AiterMLAMetadataBuilder` 把 `_decode_causal` 放在 per-ubatch 实例上、而不是扩展共享 `_build_decode` 契约的重构 (commit 5d9817e6), 是避免为 AMD 专有需求改动 CUDA 也要实现的接口的好范例; (3) `has_fp8_non_causal_qlen2_kernel` 的白名单, 是理解 AITER fp8 dispatch fold 规则的现成材料, 也是本 PR 最脆弱的假设。跟进建议: 确认 AITER 是否计划暴露能力 API; 评估在 CI 中加入 gfx942 fallback 与长 qlen 非因果块的端到端用例。

功能与动机

DSPark (DeepSeek V4.1 的投机解码 drafter) 产生的 draft query block 是非因果的: 块内 token 需要互相可见, 而不是逐 token 的因果阶梯。此前 ROCm 路径只能把这个块展平成因果形式来适配 AITER 的因果 kernel, 代价是额外的重排与无法使用 AITER 的高性能 MLA decode 路径。PR body 明确写了目标: 「Pass the real causal mask into Aiter MLA decode (`get_mla_metadata_v1` / `mla_decode_fwd`) instead of flattening a non-causal DSPark draft query block」, 并要求 「Probe `mla_decode_fwd` for a causal argument; keep Gluon and DCP verify causal-only; refuse fp8 2-token non-causal blocks until aiter folds that length」。选择 rebase 而非直接推进, 是因为原 PR #53001 与 main 冲突且作者无法向 JohnQinAMD/vllm-amd 推送。

实现拆解

1. 入口透传: 把 causal 标志接进 AITER 解码算子。`vllm/_aiter_ops.py` 中 `_rocm_aiter_mla_decode_fwd_impl` 与 `rocm_aiter_ops.mla_decode_fwd` 同时新增 `causal: bool = True`, 实现体通过 `kwargs["causal"] = causal` 与关键字参数一路传到 `torch.ops.vllm.rocm_aiter_mla_decode_fwd`。默认 True 使既有因果调用方的行为完全不变, 只有显式传 False 的 draft 路径才走新分支。
2. 能力探测: 签名 + 架构双闸门, 且失败即报错。`rocm_aiter_ops.mla_decode_supports_non_causal()` 用 `inspect.signature` 检查 `mla_decode_fwd` 是否有 `causal` 参数, 缺失时直接 `raise RuntimeError (fail-closed)`, 而不是返回 False 让上层静默回落到其它后端;

`_aiter_mla_non_causal_asm_kernels()` 再用 `on_gfx950()` 判定当前 arch 是否真的带非因果 ASM kernel; 两者在 `AiterMLABackend.supports_non_causal()` 里取与。这一层是因为 pinned AITER v0.1.21.post2 在 gfx942 上同样导出 `causal=`, 但 manifest 里没有非因果 kernel, 选中后只会打到 `cannot get heuristic kernel!`。

- 元数据构造: 把 `causal` 状态留在 per-ubatch builder 内。`AiterMLAMetadataBuilder` 新增类属性 `supports_non_causal_multi_token_decode = True` 与实例状态 `_decode_causal`, `build()` 从 `common_attn_metadata.causal` 记录, `_build_decode()` 读取;
`get_mla_metadata_v1(..., is_causal=causal)` 由原来的硬编码 `True` 改为真实值。
`persistent-metadata` 的 gate 增加 `not causal` 短路分支——非因果块必须保留 `schedule`, 因为 `schedule` 是唯一携带 mask 的载体。这样改是为了不改动
`MLACommonMetadataBuilder._build_decode` 这份 CUDA 后端也在实现的共享契约。
- 形状守卫: 拒绝 AITER 没有非因果 kernel 的组合。`_build_decode` 中, 当 `not causal and max_qo_len == 2 and fp8 cache and not AiterMLAHelper.has_fp8_non_causal_qlen2_kernel(num_heads)` 时抛 `ValueError`, 白名单
`_AITER_FP8_NON_CAUSAL_QLEN2_HEADS = (32, 64, 96, 128)` (48/80/112 会被 fold 到 H16 但保留 Q2, 无对应 kernel); `validate_configuration` 覆盖基类实现, 对 `use_non_causal and dtype == torch.float16 and kv_cache_dtype in (None, "auto", "float16")` 追加 `non-causal fp16 MLA decode not supported`, 让自动选择回落到 `TRITON_MLA`。
- 测试配套。`tests/kernels/attention/test_rocm_aiter_mla_fp8_support.py` 扩充 `_install_fake_aiter_modules` 支持 `supports_causal`, 新增
`test_missing_causal_arg_fails_closed`、`test_supports_non_causal_requires_arch_kernels`、`test_decode_fwd_forwards_causal_to_aiter`、
`test_non_causal_fp16_auto_cache_is_rejected`;
`tests/v1/attention/test_rocm_aiter_mla_mtp_split.py` 新增 `_build_non_causal` 驱动器与三组「拒绝 / 放行」用例; `tests/v1/attention/test_rocm_aiter_mla_fp8_decode_routing.py` 追加 `test_a_non_causal_block_never_routes_to_gluon`。

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 `_aiter_mla_non_causal_asm_kernels`, `AiterMLABackend.supports_non_causal`, `AiterMLABackend.validate_configuration`, `AiterMLAMetadataBuilder.supports_non_causal_multi_token_decode`): 本 PR 的核心: 新增 `_aiter_mla_non_causal_asm_kernels` 架构判定、
`AiterMLABackend.supports_non_causal` 与 `validate_configuration` 覆盖,
`AiterMLAMetadataBuilder` 增加 `supports_non_causal_multi_token_decode/_decode_causal` 并在 `build/_build_decode` 中把 mask 传给 `get_mla_metadata_v1`, 同时加上 `fp8 2-token` 守卫与 `has_fp8_non_causal_qlen2_kernel` 白名单。
- `vllm/_aiter_ops.py` (模块 算子封装; 类别 source; 类型 dependency-wiring; 符号 `_rocm_aiter_mla_decode_fwd_impl`, `rocm_aiter_ops.mla_decode_fwd`, `rocm_aiter_ops.mla_decode_supports_non_causal`): 新增 `causal` 参数在 `_rocm_aiter_mla_decode_fwd_impl / rocm_aiter_ops.mla_decode_fwd` 的透传, 以及 `mla_decode_supports_non_causal()` 这个 fail-closed 的签名探测器; 它是 `causal` 语义真

正抵达 aiter kernel 的最后一跳。

- tests/v1/attention/test_rocm_aiter_mla_mtp_split.py (模块 解码元数据测试; 类别 test; 类型 test-coverage; 符号 test_build_hands_common_causal_flag_to_aiter_builder, _build_non_causal, test_non_causal_build_hands_the_mask_to_aiter, test_a_two_token_fp8_block_is_refused_without_a_fold) : 覆盖了本 PR 最容易出错的语义: build 把 common metadata 的 causal 写进 builder、_build_decode 把 is_causal 传给 get_mla_metadata_v1, 以及 fp8 2-token 白名单的拒绝 / 放行边界 (12/16/48 拒绝、32 放行、bf16 不受影响) 。
- tests/kernels/attention/test_rocm_aiter_mla_fp8_support.py (模块 能力探测测试; 类别 test; 类型 test-coverage; 符号 mla_decode_fwd_with_causal, test_missing_causal_arg_fails_closed, test_supports_non_causal_requires_arch_kernels, test_decode_fwd_forwards_causal_to_aiter) : 验证能力探测的三条契约: 旧 aiter 缺 causal= 时 fail-closed 抛 RuntimeError、supports_non_causal 必须叠加 arch kernel 判定、解码 op 必须把 causal 原样转发给 aiter; 以及 fp16 非因果组合被 validate_configuration 拒绝。
- tests/v1/attention/test_rocm_aiter_mla_fp8_decode_routing.py (模块 后端路由测试; 类别 test; 类型 test-coverage; 符号 test_a_non_causal_block_never_routes_to_gluon) : 单条用例锁定「非因果块永不路由到 Gluon」这一约束, 防止小 head 场景下非因果 draft 被错误地交给只支持因果的 Gluon verify kernel。

关键符号: _aiter_mla_non_causal_asm_kernels,
AiterMLABackend.supports_non_causal, AiterMLABackend.validate_configuration,
AiterMLAMetadataBuilder.build, AiterMLAMetadataBuilder._build_decode,
AiterMLAHelper.has_fp8_non_causal_qlen2_kernel, _rocm_aiter_mla_decode_fwd_impl,
rocm_aiter_ops.mla_decode_supports_non_causal, rocm_aiter_ops.mla_decode_fwd,
_build_non_causal, test_supports_non_causal_requires_arch_kernels

关键源码片段

tests/v1/attention/test_rocm_aiter_mla_mtp_split.py

覆盖了本 PR 最容易出错的语义: build 把 common metadata 的 causal 写进 builder、_build_decode 把 is_causal 传给 get_mla_metadata_v1, 以及 fp8 2-token 白名单的拒绝 / 放行边界 (12/16/48 拒绝、32 放行、bf16 不受影响) 。

```
def _build_non_causal(monkeypatch, *, num_heads, kv_cache_dtype, qlen, mtp_qlen):
    """以 causal=False 驱动 _build_decode, 并交回 aiter mock 以便断言调用。"""
    get_mla_metadata_v1 = mock.MagicMock()
    monkeypatch.setitem(
        sys.modules,
        "aiter",
        SimpleNamespace(get_mla_metadata_v1=get_mla_metadata_v1),
    )
    monkeypatch.setattr(
        rocm_aiter_mla, "_expand_page_indices_kernel", _NoOpTritonKernel()
    )
```

```

builder = _builder(
    mtp_decode_qlen=mtp_qlen,
    kv_cache_dtype=kv_cache_dtype,
    num_heads=num_heads,
)
# 直接置位实例状态，等价于 build() 从 common metadata 写入的值
builder._decode_causal = False
metadata = AiterMLAMetadataBuilder._build_decode(
    builder,
    block_table_tensor=torch.arange(16, dtype=torch.int32).view(2, 8),
    seq_lens_device=torch.tensor([7, 5], dtype=torch.int32),
    max_seq_len=7,
    query_start_loc_cpu=torch.tensor([0, qlen, 2 * qlen], dtype=torch.int32),
    query_start_loc_device=torch.tensor([0, qlen, 2 * qlen], dtype=torch.int32),
    num_decode_tokens=2 * qlen,
    dcp_tot_seq_lens_device=None,
)
return metadata, get_mla_metadata_v1

```

```

def test_non_causal_build_hands_the_mask_to_aiter(monkeypatch):
    """schedule 会携带 mask，因此 is_causal 必须跟随该 block 的因果性。"""
    metadata, get_mla_metadata_v1 = _build_non_causal(
        monkeypatch, num_heads=12, kv_cache_dtype="auto", qlen=8, mtp_qlen=8
    )
    assert metadata.has_persistent_metadata
    # get_mla_metadata_v1 的第 6 个位置参数就是 is_causal
    assert get_mla_metadata_v1.call_args.args[5] is False

```

```

@pytest.mark.parametrize("num_heads", [12, 16, 48])
def test_a_two_token_fp8_block_is_refused_without_a_fold(monkeypatch, num_heads):
    """16-head（以及 padded 到 16）的 qlen=2 fp8 没有非因果 kernel。

    48/80/112（以及 33 padded 到 48）会 fold 到 H16 但保留 Q2，必须与
    原生 16-head qlen 2 一样被拒绝。
    """
    with pytest.raises(ValueError, match="2-token"):
        _build_non_causal(
            monkeypatch,
            num_heads=num_heads,
            kv_cache_dtype="fp8",
            qlen=2,
            mtp_qlen=8,
        )

```

评论区精华

本轮 review 的核心张力有三处。

其一，「静默降级」还是「直接失败」。Fangzhou-Ai 起初认为 `supports_non_causal()` 的探测不该存在：「it should always be true, i.e. AITER is always available for ROCm path, this function is not needed here」，并强调「we should guarantee our best perf path is always enabled instead of failing silently... a simple log is usually ignored by users and minor perf regression in this case is very hard to be caught」。AndreasKaratzas 倾向保留动态判定并加日志。最终 ppalanga 提出 fail-closed 版本（缺 `causal=` 就 raise `RuntimeError`），Fangzhou-Ai 回「LGTM」，分歧收敛。

其二，Python 签名探测不足以表达 per-arch kernel 表。AndreasKaratzas 在 MI350 侧验证后指出 pinned AITER `v0.1.21.post2` 在 gfx942 上仍导出 `causal=`，但该 arch 的 manifest 没有非因果 kernel，`(gqa=16, ps=1, prefill=0, causal=0, qSeqLen=4)` 会走到 `cannot get heuristic kernel!`；这直接催生了 `_aiter_mla_non_causal_asm_kernels()`。同一轮还指出两条更细的形状约束：fp8 2-token 守卫过宽（「32 heads / 2 tokens folds to the noncausal 16-head / 4-token kernel; 64, 96, and 128 heads also have supported folds」，而 48/80/112 会 native abort），以及非因果 fp16 + `kv_cache_dtype="auto"` 会以 `unsupported Q dtype:fp16` abort。三条 finding 全部被采纳并分别落到 `_AITER_FP8_NON_CAUSAL_QLEN2_HEADS`、`has_fp8_non_causal_qlen2_kernel` 与 `validate_configuration`。

其三，跨平台连带改动与测试质量。Fangzhou-Ai 质疑 `vllm/models/dots3_note/nvidia/attention.py` 里的 `causal` 参数「passed to the function then deleted immediately」，并提示 NVIDIA 侧改动需额外 review；该文件最终被整体回退，改为只让已是 Aiter 签名的后端接收 `causal=`。AndreasKaratzas 还点名测试冗余与 `__wrapped__.cache_clear()` 的写法，并在沟通层面直说：「let's not reply with AI to reviewers. I read your response twice and still dont understand what you are trying to say.」

- `supports_non_causal` 该静默降级还是直接失败 (design): 采纳 fail-closed: 签名缺失即抛错，绝不静默回落到其它 MLA 后端。Fangzhou-Ai 回复 LGTM, ppalanga 随后将该线程标记为解决。
- Python 签名探测不足以表达 per-arch kernel 表 (correctness): 新增 `_aiter_mla_non_causal_asm_kernels()` 以 `on_gfx950()` 判定架构 kernel，与签名探测在 `supports_non_causal()` 中取与；作者回复「`_aiter_mla_non_causal_asm_kernels()` now checks for gfx950」。
- fp8 2-token 非因果守卫过宽，会误杀可用配置 (correctness): 引入 `_AITER_FP8_NON_CAUSAL_QLEN2_HEADS = (32, 64, 96, 128)` 与 `has_fp8_non_causal_qlen2_kernel`，作者回复「Updated — only 32/64/96/128 pass the qlen-2 fp8 fold now」。
- 非因果 fp16 + auto cache 会 abort，需要配置级拒绝 (correctness): 在 `AiterMLABackend.validate_configuration` 中对 `use_non_causal` and `dtype == torch.float16` and `kv_cache_dtype in (None, "auto", "float16")` 追加 `non-causal fp16 MLA decode not supported`，作者回复「Updated — non-causal fp16 + auto is rejected so we fall back to Triton」。
- NVIDIA dots3_note 侧多余的 `causal` 参数 (design): 最终整体回退该文件：只有 `_build_decode` 已带 `causal` 的 Aiter 后端才接收该参数，Dots3 NOTE / FlashMLA 保持

原签名，本 PR 不再需要 NVIDIA 侧 review。

- 测试冗余与 `wrapped.cache_clear()` 写法 (testing): 去掉 `functools.cache` (`inspect` 成本低且探测本身会 `raise`)，测试收敛为 `fail-closed` / 架构 `kernel` / `decode` 透传三组；作者确认「no more `__wrapped__` now」。
- review 回复应避免 AI 生成内容 (other): 属于评审沟通过程约束，未产生代码改动，但影响了后续讨论的表达方式（作者改为在 `commit message` 与简短回复中说明改动）。

风险与影响

- 风险：
 1. 白名单与 AITER 内部 `fold` 规则强耦合。`_AITER_FP8_NON_CAUSAL_QLEN2_HEADS = (32, 64, 96, 128)` 是从 `pinned AITER v0.1.21.post2` 的 `shape dispatch` 反推出来的，`has_fp8_non_causal_qlen2_kernel` 只做集合判定。AITER 升级后若改变 `fold` 规则或补上 2-token `kernel`，这里会要么误拒绝可用配置、要么误放行并触发 `native abort`（`raise ValueError` 的注释明确写了没有 `kernel` 时是「abort the process rather than raising」）。
 2. `fail-closed` 的可用性代价。`mha_decode_supports_non_causal()` 在签名缺失时抛 `RuntimeError` 而非返回 `False`，意味着旧 `aiter` / 非镜像环境的 ROCm 服务会直接启动失败；这是刻意选择（避免静默退化到慢路径），但对用户是可观测的破坏性变更。
 3. 自动回落依赖 `validate_configuration` 的字符串列表。非因果 `fp16` 的拒绝是通过 `invalid_reasons` 驱动后端选择，属于隐式契约；若 AITER 后续修复 `fp16` 非因果，需要同步删掉这段守卫，否则能力被永久锁死。
 4. 非因果路径的验证强度不均衡。仓库内测试全部基于 `monkeypatch` 的 `fake aiter` 模块与 `mock` 的 `get_mha_metadata_v1`，真正的 `kernel` 行为只在 `MI350 / gfx950` 上人工验证；测试计划里第二轮「Original #53001 bench/accuracy plan on 8x MI355X (pinned TritonMLA vs unpinned AiterMLA)」仍是未勾选状态。非因果块的 `cuda_graph / graph replay` 覆盖也只有 `reviewer` 的本地 `harness` 证据。
 5. `persistent-metadata` 分支的行为差异。`not causal or ...` 让非因果块无条件保留 `persistent schedule`，与因果路径的门控条件不同；这段逻辑目前只有 `test_non_causal_build_hands_the_mask_to_aiter` 覆盖 `is_causal` 取值，没有覆盖长 `qlen`、`padded rank` 等边界组合。
 - 影响：用户侧：ROCm + 投机解码 (DSpark/MTP) 用户是直接受益者。作者在 `MI355X (Kimi-K3 MXFP4 + DSpark, K=6, draft KV fp8)` 上给出的 `agentic conc=1` 复现结果是 `ITL p50 6.57 ms vs SA 基线约 8 ms`、`interactivity 152 tok/s/user vs 约 124`，说明该路径从「能跑」升级为「比 `TRITON_MLA` 快」。同时，`gfx942` 与旧 `aiter` 用户会看到新的报错或回落到 `TRITON_MLA`，属于能力边界收紧。系统侧：CUDA 侧无影响——`dots3_note/nvidia/attention.py` 的改动已回退，共享的 `MLACommonMetadataBuilder` 契约未被修改。ROCm 的注意力后端选择逻辑多了一条「签名 + 架构」联合判定，为后续其它 `non-causal` 特性（如 `DCP verify`、`Gluon`）预留了模式。团队侧：这份改动把「AITER 能力靠 `inspect` + 硬编码 `arch` + 硬编码 `shape` 白名单」的现实暴露得很清楚，`_aiter_mha_native_h24_metadata_supported` 的注释里也早已吐槽「Until AITER exposes a capability API」。讨论中确立的 `fail-closed` 与「

不使用 AI 回复 reviewer] 两条文化约定，对后续 ROCm PR 有示范意义。

- 风险标记：核心解码路径变更，AITER 版本与架构强耦合，fp8 形状白名单脆弱，依赖运行时签名探测，非因果路径缺少仓库内端到端测试，验证证据集中在单卡型

关联脉络

- PR #56903 [Perf][DSpark] Collapse DeepSeek-V4.1 draft states before SP all-gather: 同属 DeepSeek V4.1 DSpark 草稿路径：那条 PR 优化草稿状态在 SP all-gather 前的通信载荷，本条则是让草稿块的非因果语义直接在 AITER kernel 上成立，两者共同构成 DSpark 在 ROCm 侧的端到端优化链条。
- PR #56387 [Bugfix][Spec Decode][MoE] Avoid uninitialized EPLB state in DeepSeek V4.1 DSpark drafter: 同样是 DSpark drafter 在 ROCm/V1 路径上的启动与状态初始化问题，与本 PR 的「草稿后端能力探测」属于同一子系统的相邻风险面。
- PR #56825 [Bugfix][SparseMLA] Fix piecewise cudagraph capture crash in index group: 同为 MLA attention backend 的路由 / 时序问题修复，涉及 decode 元数据与 cudagraph 交互；本 PR 也调整了 persistent metadata 的 gate 条件，属于同一类后端分派风险。
- PR #56969 [Bugfix] Prevent out-of-bounds access in FlashInfer SM90 sparse MLA mixed batches: 同属 MLA decode 在混合 / 非常规 batch 形状下的正确性修复，与本 PR 对「非因果多 token decode」这一非常规形状的守卫思路一致。
- PR #53001 原 Aiter MLA 非因果 decode draft 支持 PR（标题未在材料中给出）：本 PR 就是它的 rebase 重开：body 明确写「Rebase of #53001 onto current main. Original PR is conflicting and this account cannot push to JohnQinAMD/vllm-amd」，并沿用其原始描述与 MI355X 结果。