

PR #50439 完整报告

vllm-project/vllm

[Attention] Extend XQA decode support on SM90

合并时间: 2026-09-11 14:19

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50439>

执行摘要

- 一句话: FlashInfer XQA 解码扩展到 SM90
- 推荐动作: 值得精读。这是 FlashInfer 注意力后端一次「能力边界重新划线」的改动, 重点读三处: `get_cudagraph_support` 的资格判定 (`head_dim`、非因果、DCP 三个维度如何组合成 `UNIFORM_BATCH` / `UNIFORM_SINGLE_TOKEN_DECODE`)、`_compute_decode_query_lens` 的定宽 vs `varlen` 分流条件、以及 `supports_sink` 与 `_get_prefill_wrapper` 必须成对修改的耦合关系。建议合并后人工确认两点: SM100 非因果 + `sinks` 是否仍会静默丢 `sink`; SM90 + sliding window + FP8-Q 在非 0.6.17 FlashInfer 环境下是否会被静默放行。若要在 H100 上启用 XQA, 建议对比验证投机解码在 ragged query 下的数值正确性, 而不只依赖元数据单测。

功能与动机

PR body 明确这是 #49718 (为 SM12x 添加专用 FlashInfer XQA 路径) 的后续: "Use FlashInfer's dedicated XQA API on SM90 for uniform and ragged speculative decode.", 并要 "Support non-causal draft attention and attention sinks on the SM90 XQA path."。同时清理过时守卫: "Remove the obsolete SM90 sliding-window guard; current main pins FlashInfer 0.6.17 which contains the bug streamfix." (对应 [flashinfer-ai/flashinfer#3578](#))。对 SM100 则反向收紧: "Keep SM100 DCP CUDA graphs limited to uniform single-token decode because variable-length trtllm-gen decode is not graph-safe with DCP.", 并明确 "Preserve padded query/output rows for single-token full CUDA graphs while keeping multi-token ragged queries packed."。作者还声明这不是 #37592/#47306/#50885 的重复、无需升级依赖, 并已在 H100 上用 GPT-OSS 验证。

实现拆解

1. kernel 选择与符号统一 (vllm/v1/attention/backends/flashinfer.py): 把 `use_dedicated_xqa` 重命名为 `use_xqa`, 并去掉 `is_device_capability_family(120)` 这层架构门禁, 使 SM90 与 SM12x 在 `flashinfer_trtllm_api_decode_kernel == FlashInferDecodeKernel.XQA` 时走同一条专用 XQA 路径; `supports_spec_as_decode` 由 `kernel == TRTLLM_GEN or use_dedicated_xqa` 简化为 `flashinfer_trtllm_api_decode_kernel is not None`, 于是 SM90 投机解码的 `reorder_batch_threshold` 也会落到 1, 多 token ragged query 才能被收进 decode batch。新增模块级 `_is_xqa_head_dim_supported()` 把 `16 <= head_dim <= 256 and head_dim`

% 16 == 0 抽出，供初始化与 CUDA graph 判定共用，避免两处判据漂移。

2. 平台能力与 sink 生命周期对齐：supports_sink() 从「SM12x XQA 或 SM100 trtllm-gen」扩展为 SM90/SM12x 的 XQA decode 或 SM100 的 trtllm-gen prefill+decode；_get_prefill_wrapper() 中启用 BatchAttentionWithAttentionSinkWrapper 的条件同步扩为 SM90/SM12x。commit 历史显示这里经过一次反复：a52d943 先关闭 SM90 sink（理由 "Hopper XQA provides decode only, so enabling sinks advertises a prefill path that cannot consume them"），3c2d5eb 撤销该提交，2d58ccf 才通过 sink-aware native prefill wrapper 补齐 prefill 侧，让对外宣称的 sink 能力覆盖完整 attention 生命周期。
3. CUDA graph 资格重写：删除「SM90 固定返回 UNIFORM_SINGLE_TOKEN_DECODE」的分支，改为统一判定——decode_context_parallel_size > 1 直接返回单 token 均匀解码（XQA 无 LSE，变长 trtllm-gen 在 DCP 下也非 graph-safe）；has_trtllm_support 更名 has_uniform_batch_support 并追加 is_xqa_arch and not _is_xqa_head_dim_supported(spec.head_size) 的降级条件；非因果分支改为 not use_non_causal or is_xqa_arch。结果是 SM90 在 XQA 可用配置下可声明 UNIFORM_BATCH，而 head_dim=512（如 Gemma 4 全局注意力组）和 DCP 组合被挡在 UNIFORM_BATCH 之外（后者来自 CodeRabbit 发现，并在测试中补了 head_size=512 的降级用例）。
4. decode query 宽度推导分流：_compute_decode_query_lens() 按 self.use_xqa 分两条路径。trtllm-gen 路径先做「总 token 数可整除 + 每请求长度严格等于均值」的双重校验，只有完全均匀才返回定宽 q_len_per_req；否则返回 max_q_len 与 qo_indptr[:num_decodes+1]，把 padded 投机解码交给 varlen API。XQA 路径保持原逻辑：取非零长度最大值作为 query 宽度，返回 ragged 偏移与逐请求有效长度（0 表示 CUDA graph padding 请求），在 single-token full CUDA graph 下保留 padded query/output 行。
5. 清理与测试 / 配置配套：删除 SM90 sliding-window 的 NotImplementedError 守卫（依据是 main 已 pin 含上游修复的 FlashInfer 0.6.17）；get_q_data_type() 的 FP8-Q 注释按 reviewer 意见区分 SM90 native FA3 与 SM100 trtllm-gen。测试侧：
tests/v1/attention/test_attention_backends.py 新增
test_flashinfer_xqa_single_token_decode_preserves_cudagraph_padding（用 mock_xqa 捕获传给 flashinfer_xqa_batch_decode_with_kv_cache 的 query/out/block_tables 形状，断言仍是 4 行而非 2 行）和参数化
test_flashinfer_trtllm_gen_padded_decode_uses_varlen_offsets，并把
use_dedicated_xqa 断言同步为 use_xqa、cudagraph 期望改为 UNIFORM_BATCH 且补 head_size=512 用例；tests/v1/attention/test_flashinfer_dcp_spec_reorder.py 增加 DCP 下 get_cudagraph_support(...) == UNIFORM_SINGLE_TOKEN_DECODE 断言。无依赖、部署或 schema 变更。

关键文件：

- vllm/v1/attention/backends/flashinfer.py（模块 注意力后端；类别 source；类型 core-logic；符号 _is_xqa_head_dim_supported, FlashInferBackend.supports_sink, FlashInferMetadataBuilder.get_cudagraph_support, FlashInferMetadataBuilder._compute_decode_query_lens）：唯一的源码改动文件，承载

本 PR 全部核心逻辑：XQA 在 SM90 的启用（use_xqa 替换 use_dedicated_xqa）、sink 能力扩展、get_cudagraph_support 资格重写、_compute_decode_query_lens 的定宽 /varlen 分流，以及删除 SM90 sliding-window 守卫。

- tests/v1/attention/test_attention_backends.py（模块 注意力后端；类别 test；类型 test-coverage；符号 test_flashinfer_xqa_single_token_decode_preserves_cudagraph_padding, mock_xqa, test_flashinfer_trtllm_gen_padded_decode_uses_varlen_offsets, test_flashinfer_xqa_query_lens_preserve_cudagraph_padding）：新增两个关键回归用例并调整既有断言：mock_xqa 验证 single-token full CUDA graph 下 XQA 保留 padded 行；参数化用例验证 trtllm-gen 在非均匀 padded decode 下改用 varlen offsets；同时把 cudagraph 期望改为 UNIFORM_BATCH 并补 head_size=512 降级与 SM90 sink 用例。
- tests/v1/attention/test_flashinfer_dcp_spec_reorder.py（模块 注意力后端；类别 test；类型 test-coverage；符号 test_flashinfer_gqa_dcp_spec_decode_clamps_reorder_threshold）：为「DCP + 投机解码时 get_cudagraph_support 必须返回 UNIFORM_SINGLE_TOKEN_DECODE」补断言，防止后续把 DCP 重新放回 UNIFORM_BATCH 导致变长 trtllm-gen decode 进图。

关键符号：_is_xqa_head_dim_supported, FlashInferBackend.supports_sink, FlashInferMetadataBuilder.get_cudagraph_support, FlashInferMetadataBuilder._compute_decode_query_lens, FlashInferMetadataBuilder._get_prefill_wrapper, FlashInferMetadataBuilder.init

关键源码片段

vllm/v1/attention/backends/flashinfer.py

唯一的源码改动文件，承载本 PR 全部核心逻辑：XQA 在 SM90 的启用（use_xqa 替换 use_dedicated_xqa）、sink 能力扩展、get_cudagraph_support 资格重写、_compute_decode_query_lens 的定宽 /varlen 分流，以及删除 SM90 sliding-window 守卫。

vllm/v1/attention/backends/flashinfer.py（按本 PR 的新增 / 重写逻辑整理）

```
def _is_xqa_head_dim_supported(head_dim: int) -> bool:
    # 专用 XQA API 只接受 16..256 之间、且能被 16 整除的 head_dim。
    # 抽成模块级函数后，kernel 选择（__init__）与 CUDA graph 资格
    # （get_cudagraph_support）共用同一判据，避免两处漂移：
    # 初始化阶段会把不合格的 XQA kernel 清空并回退 native FlashInfer，
    # 若 graph 侧仍声称 UNIFORM_BATCH，就会为后端根本执行不了的
    # 投机解码路径捕获 graph。
    return 16 <= head_dim <= 256 and head_dim % 16 == 0
```

摘自 FlashInferMetadataBuilder，仅保留本 PR 重写的 classmethod

```
class FlashInferMetadataBuilder:
    @classmethod
    def get_cudagraph_support(
        cls,
```

```

vllm_config: VllmConfig,
kv_cache_spec: KVCacheSpec,
) -> AttentionCGSupport:
    """返回 FlashInfer 后端可用的 CUDA graph 捕获级别。"""
    # DCP 场景一律收紧为「单 token 均匀解码」：XQA 不返回 LSE,
    # 无法支撑 DCP 的 LSE 归约；而变长 trtllm-gen decode 在 DCP 下
    # 也不是 graph-safe 的，所以不做任何 UNIFORM_BATCH 承诺。
    if vllm_config.parallel_config.decode_context_parallel_size > 1:
        return AttentionCGSupport.UNIFORM_SINGLE_TOKEN_DECODE

    kv_specs = iter_layer_specs(kv_cache_spec)
    num_qo_heads = vllm_config.model_config.get_num_attention_heads(
        vllm_config.parallel_config
    )
    # 本 PR 之后 SM90 也走专用 XQA，因此把 SM90 与 SM12x 一并视为
    # 「XQA 架构」，两者的 graph 资格判定保持一致。
    is_xqa_arch = current_platform.is_device_capability(
        90
    ) or current_platform.is_device_capability_family(120)
    has_uniform_batch_support: bool = len(kv_specs) > 0
    for spec in kv_specs:
        if not isinstance(spec, AttentionSpec):
            # FlashInfer 只作用于注意力层，其他 KV 规格不参与判定。
            continue
        # 第二个条件是 CodeRabbit 抓出的补丁：XQA 架构上 head_dim
        # 不合格（例如 Gemma 4 的 head_dim=512 全局注意力组）时，
        # 初始化已经放弃 XQA，这里必须同步降级，否则会申请不存在的
        # XQA 投机解码 graph。
        if not can_use_trtllm_attention(
            num_qo_heads=num_qo_heads,
            num_kv_heads=spec.num_kv_heads,
            is_prefill=False,
        ) or (is_xqa_arch and not _is_xqa_head_dim_supported(spec.head_size)):
            has_uniform_batch_support = False
            break

    # 非因果注意力只有 XQA 架构能承载投机解码的变长 query；
    # SM100 的 trtllm-gen 非因果路径必须退回单 token 均匀解码。
    use_non_causal = vllm_config.attention_config.use_non_causal
    if has_uniform_batch_support and (not use_non_causal or is_xqa_arch):
        return AttentionCGSupport.UNIFORM_BATCH
    return AttentionCGSupport.UNIFORM_SINGLE_TOKEN_DECODE

```

评论区精华

本轮 review 的主线是「后端宣称的能力必须与实际可执行的 kernel 严格一致」，共有三条有价值交锋：

1. CUDA graph 资格漏判 head_dim (CodeRabbit, 已采纳) : "A group with head_dim=512 can return UNIFORM_BATCH here, but initialization clears the XQA kernel at Lines 823-844." 结论是在 `get_cudagraph_support` 中复用 `_is_xqa_head_dim_supported`, 并在测试里新增 head_size=512 期望 UNIFORM_SINGLE_TOKEN_DECODE 的断言。
2. 定宽 decode 分支误判 padded 投机解码 (CodeRabbit 提出, benchislett 认同) : "`num_decode_tokens % num_decodes == 0` does not prove that every request has the same query length." 反例为 `[0,2,4,4,4]/num_decodes=4/num_decode_tokens=4`。benchislett 明确回应 "The bot is right", "the suggested fix LGTM", 最终按建议改为比较 `decode_q_lens` 与候选均值, 并补了参数化回归测试。
3. SM100 非因果 sink 是否会被静默丢弃 (CodeRabbit outside-diff, 状态未确认) : 评审指出 SM100 上 `self.use_xqa` 为假、非因果时 `all_uses_trtllm` 恒为假, 守卫只在 DCP 或 cascade 时才抛错, `_get_prefill_wrapper(causal=False)` 会选到不消费 `self.sinks` 的 `BatchPrefillWithPagedKVCacheWrapper`, "Attention sinks are silently ignored." pavanimajety 要求 "make the error specific to kernel / wrapper, rather than dcp prefill / cascade path for clarity. Also please retain the suggestion.", 但最终错误文本仍是 "does not support attention sinks on the DCP prefill or cascade path.", 从现有材料看只做了部分处理。

另有一条低风险的注释纠偏 (pavanimajety: "nit: fix comment reg: SM100 + XQA"), 随后由 b1a6b5d "Correct FlashInfer FP8-Q backend rationale" 提交修正。所有讨论最终以 benchislett、pavanimajety 两位 APPROVED 收敛。

- `get_cudagraph_support` 未校验 XQA head_dim, 误报 UNIFORM_BATCH (correctness): 已采纳。新增 `is_xqa_arch` and not `_is_xqa_head_dim_supported(spec.head_size)` 降级条件, 并在 `test_attention_backends.py` 中用 head_size=512 的 `kv_cache_spec` 断言返回 UNIFORM_SINGLE_TOKEN_DECODE。
- 定宽 decode 分支误判 padded 投机解码的每请求长度 (correctness): 已采纳建议: 改为比较 `decode_q_lens` 与候选均匀长度 `q_len_per_req`, 不满足则返回 `max_q_len` 与 `qo_indptr[:num_decodes+1]` 走 `varlen`; 新增参数化回归测试 `test_flashinfer_trtllm_gen_padded_decode_uses_varlen_offsets` 覆盖 `[0,3,3]` 与 `[0,2,4,4,4]`。
- SM100 非因果 sink 路径可能静默丢弃 sinks (correctness): 部分处理: 错误信息收敛为 "FlashInfer backend does not support attention sinks on the DCP prefill or cascade path.", 但仍是 DCP/cascade 口径, 未见针对 SM100 非因果路径的显式拒止, 无法从现有材料确认是否已修复。
- SM90 sink 支持是否应与 XQA decode 一起打开 (design): 以「sink 能力必须覆盖 prefill + decode 完整生命周期」为准: `supports_sink()` 纳入 SM90, `_get_prefill_wrapper()` 的 `BatchAttentionWithAttentionSinkWrapper` 条件同步扩展为 SM90/SM12x。
- FP8-Q 注释中 SM100 与 XQA 的表述错位 (documentation): 已修正 (commit b1a6b5d "Correct FlashInfer FP8-Q backend rationale"), 注释改为区分 SM90 native FA3 prefill 与 SM90/SM12x XQA decode、SM100 trtllm-gen。

- CI、冲突与预提交的合并流程摩擦 (other): 流程性讨论, 最终 Buildingkite CI 全绿并由 benchislett、pavanmajety APPROVED 合并; diff 中含合并解冲突噪声, 阅读时需区分语义变更。

风险与影响

- 风险: 1. SM100 非因果 sink 路径可能静默丢弃 sinks (未闭合, 最高风险):
CodeRabbit 指出在 SM100 上 `self.use_xqa` 为假、非因果时 `all_uses_trtllm` 恒为假, 因此 `sinks_would_be_dropped = self.use_dcp or use_cascade` 这层守卫不会触发; `_get_prefill_wrapper(causal=False)` 会选到不接收 `self.sinks` 的 `BatchPrefillWithPagedKVCacheWrapper`, sink 被静默忽略, 表现为数值偏差而非报错。pavanmajety 要求把错误信息写具体, 但最终文本仍指向 "DCP prefill or cascade path", 无法从提供的材料确认拒止逻辑是否补齐。
- 2. 删除 SM90 sliding-window 守卫依赖外部 FlashInfer 版本: 移除该守卫的依据是 main 已 pin 含上游修复的 FlashInfer 0.6.17 (flashinfer-ai/flashinfer#3578)。若用户使用自编译或回退版本的 FlashInfer, SM90 + sliding window + FP8-Q 会从「fail-fast 明确报错」变成「静默进入 kernel」, 属于隐蔽的行为回退。
- 3. SM90 CUDA graph 资格放宽: `is_xqa_arch` 纳入 SM90 后, SM90 在 head 数量与 head_dim 均合格的条件下可声明 `UNIFORM_BATCH`。资格判定是 CPU 侧按 KV cache spec 推断的静态结论, 一旦某些模型的实际 layer spec 与运行时选中的 kernel 不一致, graph capture 失败或 replay 结果错乱的影响面比改动前 (SM90 一律单 token) 更大; PR body 仅提到 GPT-OSS on H100 的手工验证。
- 4. 每步 decode 的 CPU 侧新增比较开销: `_compute_decode_query_lens` 的 trtllm-gen 分支现在对 `qo_indptr_cpu` 做逐请求比较与 `max()`。因为对象是 CPU 张量, 不引入 GPU 同步, 预期开销可忽略, 但后续实现若把比较搬到 device 张量上会直接把同步成本打进 decode hot path。
- 5. 测试覆盖以 mock 为主: 新增用例在 CPU 上用 `object.__new__` 绕过 `__init__` 并 monkeypatch kernel 函数, 验证的是元数据推导与形状契约, 不覆盖真实 kernel 数值; SM90 端到端正确性依赖 CI 中的 Hopper 机器与放宽后的 `test_flashinfer_xqa_decode_correctness`。
- 6. 合并噪声: PR 有 11 个 commit、两次 `Merge branch 'main'` 与 mergify 报告的冲突 / 预提交失败, 最终 diff 中含合并解冲突产生的非功能性改动, review 时需区分语义变更与噪声。
- 影响: 对用户: H100 (SM90) 上使用 FlashInfer 后端的用户, decode 会自动切到专用 XQA 路径, 从原来只能单 token 均匀解码扩展到支持均匀与 ragged 投机解码, 并首次获得 attention sink 支持 (GPT-OSS 这类模型受益); 带 sliding window 的 FP8-Q 模型在 SM90 上不再被直接拒绝。SM100 + DCP 用户则被明确限制为单 token 均匀解码, 可能损失图内变长投机解码的吞吐收益, 这是有意的正确性取舍。SM12x 行为按 PR body 声明保持不变, padded single-token XQA 修正对两种架构同时生效。

对系统: `reorder_batch_threshold`、`get_cudagraph_support`、`_compute_decode_query_lens` 三者现在构成一组强耦合契约, 任一处放宽都会直接改变后端在 FULL CUDA graph 下的行为; `_is_xqa_head_dim_supported` 与 `use_xqa` 则成为 kernel

选择的单一事实来源，后续新增架构（如 SM100 XQA）必须同时更新这两处与测试。

对团队：本 PR 的 review 集中在「注释 / 错误信息必须与实际可执行路径对应」，说明该后端的能力描述已被当作对外契约看待；维护者后续改动需要同步更新用户可见的告警与报错文案，否则容易再次出现「宣称支持但实际回退」的问题。

- 风险标记：核心解码路径变更，CUDA graph 资格判定重写，SM100 sink 可能静默丢弃，删除 SM90 sliding-window 守卫，测试以 mock 为主，diff 含合并噪声

关联脉络

- PR #49718 Add dedicated FlashInfer XQA path for SM12x: 本 PR 的直接前置（标题按 PR body 描述还原）：引入专用 XQA API、FlashInferDecodeKernel.XQA 与 use_dedicated_xqa 判定；本 PR 在其之上把 XQA 扩展到 SM90 并把符号统一为 use_xqa。
- PR #50885 Native FlashInfer SM120 path (per PR body): PR body 明确 "#50885 targets the native FlashInfer SM120 path rather than dedicated SM90 XQA", 属于互补关系；二者共享 supports_sink 与 _get_prefill_wrapper 中的 SM12x 分支，需保持行为一致。
- PR #47306 Legacy shared XQA API support (per PR body): PR body 声明本 PR 不是 #37592/#47306 的重复：这些草稿基于更老的 FlashInfer/base 版本与 legacy 共享 XQA API，本 PR 走 FlashInfer 0.6.17 的专用 API，可作为前两者的替代实现。
- PR #37592 Legacy shared XQA API support (per PR body): 同上：PR body 明确排除重复，说明这是同一功能线（在更老基线上做 XQA decode）的迭代，最终以本 PR 的专用 API 方案落地。
- PR #54889 [DCP][Kernel][Perf] Fuse the empty-shard LSE mask into the A2A pack kernel: 同属 DCP 注意力路径的近期演进（DCP kernel 与 CUDA graph 相关）；本 PR 把 DCP 下的 CUDA graph 策略收紧为单 token 均匀解码，与 DCP 优化方向互为约束，属弱关联而非直接依赖。