PR #28394 完整报告

sgl-project/sglang

Upgrade fa3 hash

合并时间：2026-06-18 04:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28394

执行摘要

- 一句话：升级 FA3 hash 并支持 only_qv 模式
- 推荐动作：推荐阅读 sgl-kernel/python/sgl_kernel/flash_attn.py 中 only_qv 的实现，了解如何在不改变内核 API 的情况下通过填充空张量跳过计算路径。该设计模式在类似场景（如条件性跳过某些 matmul）中有参考价值。同时，构建配置的智能化（根据平台开关 FA3）值得在其他 CUDA 特性中推广。

功能与动机

升级 Flash Attention 3 的 hash 以获取上游 bug 修复和性能改进（特别是 PR#28425 中的 only_qv 支持）。only_qv 功能针对 QK RoPE 维度为零的注意力层（如某些模型内的特定层）可跳过不必要的 K 矩阵乘法，减少计算量。ARM 平台上默认禁用 FA3 是因为编译兼容性问题（aarch64 下 FA3 依赖的某些 CUDA 特性不可用）。

实现拆解

1. 构建配置更新（sgl-kernel/CMakeLists.txt）：更新 sgl-attn 下载 URL 的 commit hash，并根据 CMAKE_SYSTEM_PROCESSOR 和 CUDA 版本自动设置 SGL_KERNEL_ENABLE_FA3 默认值——CUDA 12.4+ 且非 ARM 时启用，ARM 默认关闭。
2. 核心 Python API 改造（sgl-kernel/python/sgl_kernel/flash_attn.py）：在 flash_attn_with_kvcache 函数中添加 only_qv 参数。当 only_qv=True 且 k_cache 为 None 时，动态创建一个空的占位张量，从而在底层跳过 K 矩阵乘法。调整 softmax_scale 计算分支，使其仅基于 qv 的维度。
3. JIT 内核封装同步（python/sglang/jit_kernel/flash_attention_v3.py 和 flash_attention.py）：将 only_qv 参数透传到 _call_fa3_kernel 中，并在 k_cache 可能为 None 时跳过 stride 断言，避免不必要的错误。
4. C++ 层接口扩展（sgl-kernel/include/sgl_flash_kernel_ops.h 和 sgl-kernel/csrc/flash_extension.cc）：在 mha_fwd 声明和 Torch 扩展注册中添加 sparse_mask_fine 和 only_qv 参数，保证底层 kernel 接口一致。

关键文件：

- sgl-kernel/python/sgl_kernel/flash_attn.py（模块 内核 API；类别 source；类型 core-logic；符号 flash_attn_with_kvcache, flash_attn_varlen_func）：核心 Python API，实现了 only_qv 逻辑，包括 k_cache 为 None 时的占位创建和 softmax_scale 的适配。

- python/sglang/jit_kernel/flash_attention_v3.py（模块 JIT 内核；类别 source；类型 core-logic；符号 flash_attn_with_kvcache, flash_attn_varlen_func）：FA3 的 JIT Kernel 包装，需要透传 only_qv 并调整 k_cache 的断言逻辑。
- sgl-kernel/CMakeLists.txt（模块 构建脚本；类别 config；类型 configuration）：构建配置核心改动：升级 FA3 依赖 hash，并根据平台决定默认是否启用。
- sgl-kernel/include/sgl_flash_kernel_ops.h（模块 C++ 接口；类别 source；类型 core-logic；符号 mha_fwd）：C++ 层 mha_fwd 签名增加 sparse_mask_fine 和 only_qv 参数。
- sgl-kernel/csrc/flash_extension.cc（模块 扩展注册；类别 source；类型 core-logic；符号 fwd）：Torch 扩展注册添加新的输入参数以匹配更新后的 FA3 接口。
- python/sglang/jit_kernel/flash_attention.py（模块 JIT 封装；类别 source；类型 core-logic；符号 flash_attn_with_kvcache）：通用 JIT 内核封装，需要将 only_qv 参数传递到 v3 实现。

关键符号：flash_attn_with_kvcache, flash_attn_varlen_func, mha_fwd, fwd

关键源码片段

sgl-kernel/python/sgl_kernel/flash_attn.py

核心 Python API，实现了 only_qv 逻辑，包括 k_cache 为 None 时的占位创建和 softmax_scale 的适配。

```
def flash_attn_with_kvcache(
    q, k_cache, v_cache, k=None, v=None, qv=None,
    ...
    only_qv=False, # 跳过 K 矩阵乘法，仅使用 QV（要求 qv 参数）
    ...
):
    # 仅支持 sm90+ 平台
    ...
    if v_cache is None:
        raise ValueError("v_cache must be provided")
    assert v_cache.stride(-1) == 1, "v_cache must have contiguous last dimension"

    # 当 only_qv=True 时，允许 k_cache 为 None，创建一个空张量占位
    if k_cache is None:
        if not only_qv:
            raise ValueError("k_cache can only be None when only_qv=True")
        # 从 q 或 k 或 fallback 推断头部大小
        if q is not None:
            k_head_size = q.shape[-1]
            k_dtype = q.dtype
            k_device = q.device
        elif k is not None:
            k_head_size = k.shape[-1]
            k_dtype = k.dtype
            k_device = k.device
```

```
        else:
            # Fallback: only_qv 内核忽略 K 实际值，一个小的占位即可
            k_head_size = 64
            k_dtype = v_cache.dtype
            k_device = v_cache.device
        k_shape = (*v_cache.shape[:-1], k_head_size)
        k_cache = torch.empty(k_shape, dtype=k_dtype, device=k_device)
        assert k_cache.stride(-1) == 1, "k_cache must have contiguous last dimension"

    # 若 q 为 None（仅当 only_qv=True），同样创建占位
    if q is None:
        if not only_qv:
            raise ValueError("q can only be None when only_qv=True")
        if qv is None:
            raise ValueError("q must be provided unless qv is provided with only_qv=True")
        q_shape = (*qv.shape[:-1], k_cache.shape[-1])
        q = torch.empty(q_shape, dtype=qv.dtype, device=qv.device)

    if softmax_scale is None:
        if only_qv:
            if qv is None:
                raise ValueError("only_qv=True requires qv to be provided")
            # 只基于 qv 的 headdim 计算 scale
            softmax_scale = (qv.shape[-1]) ** (-0.5)
        else:
            # 正常 scale 计算：考虑 q 和可选的 qv
            softmax_scale = (q.shape[-1] + (qv.shape[-1] if qv is not None else 0)) ** (-0.5)

    ... # 后续调用底层 kernel
    out, softmax_lse, *rest = _call_fa3_kernel(
        _load_fa3_kernels()["flash_attn_with_kvcache"],
        q, k_cache, v_cache, k, v, qv,
        ...,
        None, # sparse_mask_fine
        only_qv, # only_qv
    )
    return (out, softmax_lse, *rest) if return_softmax_lse else out
```

python/sglang/jit_kernel/flash_attention_v3.py

FA3 的 JIT Kernel 包装，需要透传 only_qv 并调整 k_cache 的断言逻辑。

```
@debug_kernel_api
def flash_attn_with_kvcache(
    q, k_cache, v_cache, k=None, v=None, qv=None,
    ...
    only_qv=False, # Skip K matmul when qk rope dim is 0 (requires qv)
    ...
):
    if not _is_fa3_supported():
```

```
            raise NotImplementedError(
                "flash_attn at sgl-kernel is only supported on sm90 and above"
            )

        # 当 only_qv=True 时，k_cache 可能为 None（由 sgl-kernel 内部创建占位），暂时跳过断言
        if k_cache is not None:
            assert k_cache.stride(-1) == 1, "k_cache must have contiguous last dimension"
        assert v_cache.stride(-1) == 1, "v_cache must have contiguous last dimension"

        return _call_fa3_kernel(
            _load_fa3_kernels()["flash_attn_with_kvcache"],
            q, k_cache, v_cache, k, v, qv,
            ...,
            only_qv=only_qv,
            return_softmax_lse=return_softmax_lse,
            sinks=sinks,
            out=out,
        )
```

评论区精华

本 PR 未产生公开的 review 评论，但从 commit 历史可见一次专门的签名修复提交（[FA3] Fix only_qv signature placement to match pinned sgl-attn (#28425)），表明 only_qv 参数在 Python 和 C++ 链路中的位置对齐需要协商调整。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 新增代码路径风险：only_qv=True 且 k_cache 为 None 时创建的占位张量增加了内存分配，且可能掩盖底层 kernel 的真实行为，需要充分测试。
  - FA3 依赖新 commit：新 hash 对应的上游代码可能引入回退或冲突，需关注性能与精度。
  - ARM默认禁用：ARM用户若不手动启用FA3，将自动回退到FA2路径，可能影响性能。
  - 接口兼容性：only_qv 参数新增至多个调用栈，与外部集成时需确保老版本调用方不传递意外参数。
  - 影响：影响范围：中等。只影响启用了 FA3 kernel 的场景（NVIDIA sm90+）。用户可通过 only_qv=True 对特定注意力层启用优化，降低 QK RoPE=0 时的计算量。ARM 用户需明确知道需要手动设置 SGL_KERNEL_ENABLE_FA3=ON 才能使用 FA3。影响程度：功能新增，不破坏已有使用方式（only_qv 默认为 False）。
  - 风险标记：新功能路径，ARM 平台默认禁用，依赖外部 commit，接口兼容性

关联脉络

- PR #28425 [FA3] Fix only_qv signature placement to match pinned sgl-attn: 直接关联：本 PR 中包含了 #28425 的提交，用于修复 only_qv 参数位置。