

PR #47416 完整报告

vllm-project/vllm

[perf]Add fused Kimi image preprocessing

合并时间: 2026-07-06 23:46

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47416>

执行摘要

此 PR 通过引入融合的 CPU 图像预处理路径，显著加速了 Kimi K2.5/K2.6 模型的视觉预处理，最高加速 10.242x。核心是一个基于 numba 并行化的处理器，通过查找表融合填充、归一化和分块操作。当 numba 不可用时自动回退到远程 HF 处理器。修改了 5 个文件，新增 402 行代码。

功能与动机

Kimi K2.5/K2.6 模型处理高分辨率图像（如 8×2048×2048）时，现有的远程 HF 图像预处理成为瓶颈，增加了首 token 延迟。此 PR 旨在通过本地 CPU 优化减少预处理时间。作者在 PR 描述中提供了详细基准：

```
origin processor: mean 1315.969 ms OpenCV LUT fused: mean 128.490 ms
(10.242x speedup)
```

实现拆解

1. 新增融合图像处理器(vllm/transformers_utils/processors/kimi_k25_vision_fused.py): 实现 KimiK25FusedVisionProcessor，继承 BaseImageProcessor。核心函数 `_write_fused_patches` 使用 `@njit(parallel=True)` 并行化循环，通过 `uint8→float32` 查找表进行归一化，避免了重复计算。
2. 添加 numba 可用性检查(vllm/transformers_utils/import_utils.py): 定义 `is_numba_available()` 函数，方便其他模块复用。
3. 修改图像处理器工厂(vllm/transformers_utils/processor.py): `get_image_processor` 增加 `processor_cls_overrides` 参数，允许调用者覆盖默认的处理类。
4. 集成到模型(vllm/model_executor/models/kimi_k25.py): 在 `KimiK25ProcessingInfo.__init__` 中检查 numba 可用性，若可用则将 `KimiK25FusedVisionProcessor` 作为 `processor_cls_overrides` 传入，否则使用默认的 HF 处理器。
5. 确保 fork 安全(vllm/transformers_utils/jit_monitor.py): 添加 `numba_workqueue_threading_layer` 上下文管理器，在并行 numba 调用前切换到 workqueue 线程层，避免 OpenMP 运行时在 fork 后崩溃。

以下为 `_write_fused_patches` 函数的关键实现，展示了融合操作和查找表使用：

```

if is_numba_available():
    from numba import njit, prange

    @njit(parallel=True, cache=True)
    def _write_fused_patches(
        frames: np.ndarray,
        out: np.ndarray,
        out_offset: int,
        new_h: int,
        new_w: int,
        padded_h: int,
        padded_w: int,
        patch_size: int,
        normalize_lut: np.ndarray,
    ) -> None:
        t_size = frames.shape[0]
        patch_h = padded_h // patch_size
        patch_w = padded_w // patch_size
        total = t_size * padded_h * padded_w * 3
        hwc = padded_h * padded_w * 3
        wc = padded_w * 3

        for linear in prange(total):
            t = linear // hwc
            rem = linear - t * hwc
            y = rem // wc
            rem = rem - y * wc
            x = rem // 3
            c = rem - x * 3

            value = frames[t, y, x, c] if y < new_h and x < new_w else 0

            patch_idx = (
                out_offset
                + t * patch_h * patch_w
                + (y // patch_size) * patch_w
                + (x // patch_size)
            )
            out[patch_idx, c, y % patch_size, x % patch_size] = normalize_lut[value, c]
        else:
            def _write_fused_patches(*args, **kwargs):
                raise RuntimeError("numba is required for fused Kimi image preprocessing")

```

此外，[numba_workqueue_threading_layer](#) 上下文管理器确保 numba 并行调用的 fork 安全性，其实现位于 [vllm/utils/jit_monitor.py](#)。

评论区精华

- [Isotr0py](#)建议移除环境变量，自动检测 numba：

"I think there's no need to introduce env var. We can enable numba processor when numba available, and fall back to normal ones when numba unavailable." 作者采纳，删除了 `VLLM_KIMI_K25_FUSED_IMAGE_PREPROCESS`。

- Isotr0py建议使用查找表：

"Can we fuse the normalize computation through a 8-bit precomputed lookup tables / cache as well?" 作者实现后获得 1.094x 额外加速。

- Isotr0py建议采用 `processor_cls_overrides` 模式：

"Actually, you can modify `get_image_processor` like `get_video_processor` with `processor_cls_overrides`" 作者参照重构了集成代码。

风险与影响

- 风险：numba 为可选依赖，若未安装则无加速；numba 默认线程层 omp 在 fork 后不安全，已通过 workqueue 缓解；查找表假设输入 uint8，在合法输入下安全；数值误差在 1e-6 以下。
- 影响：对 Kimi 模型用户透明加速，最高 10x；对其他模型无影响；团队可复用两个新工具函数。

关联脉络

此 PR 是 Kimi K2.5/K2.6 模型性能优化系列的一部分，同一个开发者（Isotr0py）也在其他 PR 中贡献了相关优化（如 [#47474](#) DeepSeek V4 注意力性能改进）。未来可能涉及更多模型加载和预处理优化。