PR #35336 完整报告

sgl-project/sglang

VLM: feed the packed qkv projection output to vision backends uncopied

合并时间：2026-08-19 14:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35336

执行摘要

- 一句话：视觉注意力直通 strided qkv 视图，省 HBM 拷贝
- 推荐动作：值得精读。核心看点：① 用『视图直通』替代『dense 复制』的共享内存布局优化，依赖 kernel 显式 stride 读取的接口契约；② 白名单 + opt-out 门控把『谁能吃 strided』与『谁必须吃 dense』的决策集中在一处，并逐一给出物理理由；③ 用 _RecordingBackend 观察层与后端之间的张量契约，比只断言数值更贴近接口语义；④ 对未验证 kernel 一律保守排除的工程态度。建议后续关注是否有 GPU-bound 的端到端基准来收口收益结论。

功能与动机

qkv_proj 把 q/k/v 交织写进同一 block buffer，切分后的视图只有 token 维非单位 stride；原实现对 q/k/v 各调用一次 .contiguous()，等于把整份投影输出在 HBM 里搬运两遍。以 PaddleOCR-VL 页面（10764 patches、16 heads、head_dim 72、bf16）为例，27 层每请求产生 3.74 GiB 的 HBM 流量。支持显式 stride 的后端可以直接读取这些视图，因此拷贝是纯浪费。

实现拆解

1. 在 python/sglang/srt/layers/attention/vision.py 中紧随 QKV_BACKEND_IMPL 定义新增 STRIDED_QKV_BACKENDS 白名单（fa3、fa4、triton_attn、amx_attn）。收录条件是后端通过显式 per-dim stride 读取 q/k/v；flashinfer_cudnn 因 packed element indptrs 假设 dense 布局被排除，sdpa、ascend_attn、aiter_attn、xpu_attn 因 kernel 未验证对非单位 token stride 的读取而被保守排除。
2. 在 VisionAttention.__init__中计算 self.pass_strided_qkv 门控：use_qkv_parallel && 后端在白名单 && 无 qk_normalization && 无 customized_position_embedding_applier。qk_normalization（internvl）经 q.view(N, -1, embed) 原地归一化 q/k，kernel 按 dense 步幅写回会污染共享 buffer 的 k/v 区域；模型 applier（如 mllama4）会随意 reshape q/k，两者必须拿到 dense 输入。qk_normalization_by_head_size（GLM-OCR）无需 opt-out，因其内部 reshape(-1, head_size) 自行物化 dense 副本。GQA 无需特例：qkv.view(tokens, 3, head, head_dim)[:, i] 与 split + reshape 字节级一致。
3. 在 forward 的 use_qkv_parallel 分支中，将原 if not (_is_cpu and _is_cpu_amx_available): 谓词替换为 if not self.pass_strided_qkv:，仅在该条件成立时才执行三次 .contiguous()。amx_attn 在白名单内，CPU-AMX 默认路径行为不变。

4. 新增 CPU 单元测试 test/registered/unit/layers/attention/test_vision_strided_qkv.py：gloo 单 rank 进程组 fixture、monkeypatch _determine_attention_backend 绕过平台检测、_RecordingBackend 记录层递给后端的张量。门控矩阵参数化覆盖 8 个后端 + 3 个变体；数值等价性测试验证 strided 视图与 dense 复制字节级同值（MHA 与 GQA）且层输出 bit-identical。
5. 验证配套：8 种形状（含 GQA、token 数 1023、fp16）下 flash_attn_varlen_func 与 Triton context_attention_fwd 输出 bit-identical；PaddleOCR-VL、Qwen2.5-VL-3B、Qwen3-VL-2B 端到端 greedy 输出 byte-identical；H200 fa3 下 encoder 66.42→64.88 ms（-2.3%）、单层 -7.0%、峰值显存每层 -72 MiB。端到端吞吐在噪声带内，PR 自认 CPU-bound harness 无法判定 GPU 侧收益。

关键文件：

- python/sglang/srt/layers/attention/vision.py（模块 注意力层；类别 source；类型 core-logic；符号 STRIDED_QKV_BACKENDS, VisionAttention.init, VisionAttention.forward）：核心变更文件：新增 STRIDED_QKV_BACKENDS 白名单、pass_strided_qkv 门控，并替换 forward 中无条件 contiguous 的分支，是本次性能优化的主路径。
- test/registered/unit/layers/attention/test_vision_strided_qkv.py（模块 单元测试；类别 test；类型 test-coverage；符号 _RecordingBackend, gloo_world, single_rank, _build）：新增 CPU 单元测试，覆盖 8 后端 × 3 变体的门控矩阵、strided 视图与 dense 复制的字节级等值（MHA 与 GQA），是本次优化正确性的主要防线。

关键符号：VisionAttention.init, VisionAttention.forward, test_pass_strided_qkv_gating, test_strided_qkv_carries_the_dense_values

关键源码片段

python/sglang/srt/layers/attention/vision.py

核心变更文件：新增 STRIDED_QKV_BACKENDS 白名单、pass_strided_qkv 门控，并替换 forward 中无条件 contiguous 的分支，是本次性能优化的主路径。

```
# 只有按显式 per-dim stride 读取 q/k/v 的后端，才可直接消费 qkv_proj
# 打包输出的 strided 视图。flashinfer_cudnn 的 packed element indptrs 假设
# dense 布局；sdpa、ascend_attn、aiter_attn、xpu_attn 的 kernel 未验证对
# 非单位 token stride 的读取，均不列入白名单。
STRIDED_QKV_BACKENDS = frozenset({"fa3", "fa4", "triton_attn", "amx_attn"})
```

```
# __init__ 中：qkv_proj 将 q/k/v 交织写入同一块 buffer，切分出的视图只有
# token 维是非单位 stride。pass_strided_qkv 为真时，后端直接接收视图，
# 省去每层三次 HBM 拷贝。两个中间消费者需要 dense 布局，故 opt-out：
# - internvl 的 qk_normalization 经 q.view(N, -1, embed) 原地归一化，
# 按 dense 步幅写回会污染共享 buffer 的 k/v 区域；
# - 模型自定义位置编码 applier 会随意 reshape q/k（如 mllama4）。
# qk_normalization_by_head_size 无需 opt-out：其 reshape(-1, head_size)
# 会自行物化 dense 副本。
```

```
self.pass_strided_qkv = (
    use_qkv_parallel
    and self.qkv_backend_name in STRIDED_QKV_BACKENDS
    and not qk_normalization
    and customized_position_embedding_applier is None
)

# forward 的 use_qkv_parallel 分支：仍从同一 buffer 切出 q/k/v，
# 视图只有 token 维 stride 非 1
dqkv, _ = self.qkv_proj(x)
q, k, v = qkv.split([self.q_size, self.kv_size, self.kv_size], dim=-1)
q = q.reshape(bsz * s, num_heads, -1) # 视图，仍保持 strided
k = k.reshape(bsz * s, num_kv_heads, -1)
v = v.reshape(bsz * s, num_kv_heads, -1)
# …（rearrange 等布局整理省略，注意力计算语义不变）…
# 仅白名单外后端需要 dense 化；白名单内后端直接读取视图，
# 避免整份投影输出在 HBM 中往返搬运
if not self.pass_strided_qkv:
    q = q.contiguous()
    k = k.contiguous()
    v = v.contiguous()
```

test/registered/unit/layers/attention/test_vision_strided_qkv.py

新增 CPU 单元测试，覆盖 8 后端 × 3 变体的门控矩阵、strided 视图与 dense 复制的字节级等值（MHA 与 GQA），是本次优化正确性的主要防线。

```
# 门控矩阵：8 个后端 + 3 个 qk-norm / applier 变体。
# expected 为 True 表示该配置下 pass_strided_qkv 成立：
# 后端能吃 strided 视图，且中间没有需要 dense 布局的消费者。
@pytest.mark.parametrize(
    ("backend", "extra", "expected"),
    [
        # 按显式 per-dim stride 读取，可直通
        ("fa3", {}, True),
        ("fa4", {}, True),
        ("triton_attn", {}, True),
        ("amx_attn", {}, True),
        # kernel 未验证对非单位 token stride 的读取，保守排除
        ("sdpa", {}, False),
        ("flashinfer_cudnn", {}, False),
        ("xpu_attn", {}, False),
        ("aiter_attn", {}, False),
        # internvl 通过视图原地归一化 q/k，假设 dense 布局
        ("fa3", {"qk_normalization": True}, False),
        # 模型 applier 拥有 q/k 布局，必须拿到 dense 输入
        ("fa3", {"customized_position_embedding_applier": lambda *args: None}, False),
        # 该变体自行 reshape 出 dense 副本，视图无妨
        ("fa3", {"qk_normalization_by_head_size": True}, True),
    ],
```

```
)
def test_pass_strided_qkv_gating(monkeypatch, single_rank, backend, extra, expected):
    # 用记录型后端替换真实实现，只观察层传给后端的张量契约
    monkeypatch.setitem(vision.QKV_BACKEND_IMPL, backend, _RecordingBackend)
    assert _build(backend, **extra).pass_strided_qkv is expected
```

评论区精华

本 PR 无 review 评论，但 PR body 对设计取舍作了详细论证，相当于自我答辩：

- flashinfer_cudnn 排除理由：其 packed element indptrs 假设 dense 布局，不能吃 strided 视图。
- 未验证 kernel 的后端（sdpa、ascend_attn、aiter_attn、xpu_attn）一律保守排除，避免静默错误。
- qk_normalization 的原地写风险：apply_qk_norm 通过 q.view(N, -1, embed) 原地归一化，kernel 按 dense 步幅写回会破坏共享 buffer 的 k/v 区域。
- qk_normalization_by_head_size 不需要 opt-out：其 reshape(-1, head_size) 自己物化 dense 副本。
- GQA 等价性论证：view 切分与 split+reshape 字节一致，k/v slice 的 stride 形状相同。
- 对收益的诚实声明：No end-to-end win is claimed yet，端到端基准仍在噪声带内。
- 暂无高价值评论线程

风险与影响

- 风险：正确性风险：STRIDED_QKV_BACKENDS 是静态字符串白名单，若某后端 kernel 实现被替换为按 dense 假设读取，会在白名单内静默产生错误结果；新增后端时必须人工评估其 stride 读取能力并补测试。数据污染风险：qk_normalization 的 opt-out 若遗漏，k/v 区域会被原地归一化写回污染，属于静默精度错误，当前仅由单元测试覆盖已知消费者。性能风险：端到端收益未证实，吞吐基准中 GPU 利用率均值仅 17.4%（CPU-bound on image preprocessing），无法分辨 GPU 侧节省；声称收益仅限 encoder 级。兼容性风险：替换 _is_cpu and _is_cpu_amx_available 谓词后，amx_attn 白名单保持 CPU-AMX 默认路径不变，其余后端行为语义等价（原全拷贝，现白名单内不拷贝但数值 bit-identical）。
- 影响：用户面：PaddleOCR-VL、Qwen2.5-VL、Qwen3-VL、InternVL 等 VLM 视觉编码器在 fa3/fa4/triton/AMX 平台上每层峰值显存下降约 72 MiB、单层延迟约 -7%；长序列、高分辨率 patch 场景收益更明显，端到端延迟暂在噪声带内。系统面：NVIDIA（fa3/fa4）、Triton、CPU-AMX 平台自动受益，其余后端维持原 dense 路径，无行为变化。团队面：确立了『后端必须声明 stride 读取能力』的接口契约，后续新增 vision attention 后端需评估是否可加入白名单；测试中的 gloo 单 rank fixture 与 _RecordingBackend 观察模式可复用于其他层到后端契约的测试。
- 风险标记：注意力核心路径变更，依赖后端 stride 读取约定，端到端收益未验证

关联脉络

- PR #35318 [Perf] PaddleOCR-VL: overlap page preprocessing, pack the ViT, enable prefill CUDA graph: 本 PR 的性能数据几乎全部基于 PaddleOCR-VL SigLIP 编码器，二者共享 VisionAttention 路径，属于同一条 VLM 编码器性能优化链条。
- PR #12961 Fix DP attention on CPU: 该 PR 涉及 amx_attn 与 _is_cpu and _is_cpu_amx_available 谓词；本 PR 用 pass_strided_qkv 替换该谓词，并把 amx_attn 放进白名单以保持 CPU-AMX 默认路径不变。