PR #29533 完整报告

sgl-project/sglang

feat(mem_cache): page-major (layer-major within a page) KV/state layout

合并时间：2026-06-30 05:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29533

执行摘要

- 一句话：新增可选的页面主序 KV 布局，为页面粒度操作奠基
- 推荐动作：此 PR 是 KV 缓存架构调整的基石，推荐精读。关键设计决策包括：利用 constexpr 折叠实现零开销切换；使用子类化替代条件分支来区分布局；对不兼容方法主动抛出 NotImplementedError 防止误用。build_page_major_mha_views 的 strided view 构建值得学习。

功能与动机

SGLang 原有的 KV 缓存按层独立张量存储（layer-major），每个 token/page 分散在 num_layers 个独立分配中。本 PR 新增可选的物理布局 --enable-page-major-kv-layout，将最外层轴改为页面：每个页面的全部深度（所有层的 K/V 和 Mamba 状态）在连续字节缓冲区中按页面内层序排列。将页面深度共置是为页面粒度 KV 操作（移动、传输、卸载、分配）打基础，并改善这些路径的局部性。

实现拆解

1. 新增布局构建器（mem_cache/layout/page_major.py）：定义 mha_entry_bytes、build_page_major_mha_views、mamba_entry_bytes、build_page_major_mamba_views 等纯函数，基于 torch.as_strided 在原始 uint8 缓冲区上构建每层的 stride view，不持有分配器状态。
2. 集成到内存池（mem_cache/memory_pool.py）：新增 PageMajorMHATokenToKVPool 子类通过 _store_kv_layer 和 _move_kv_cache_impl 模板钩子实现布局感知的 KV 存储 / 移动；MambaPool 添加信封分支；布局不兼容的方法（如 get_contiguous_buf_infos）抛出 NotImplementedError 防止静默错误。
3. 修改 Triton 内核（triton_ops/cache_move.py、decode_attention.py、extend_attention.py）：为 store_cache_4d 和 attention 内核增加 page_size 参数，通过 constexpr 确保 page_size=1 时生成与旧布局完全相同的 SASS。
4. 配置校验（server_args.py）：添加 --enable-page-major-kv-layout 标志和 _handle_page_major_kv_layout 验证器，强制要求所有 attention/linear-attn/Mamba backend 为 Triton。
5. 测试套件：新增 CPU 端布局视图测试、Triton 内核 byte-identical 测试（decode/extend/store_cache_4d）、端到端 GSM8K 精度测试（gpt-oss 和 qwen-hybrid）

并在 CI 注册。GDN prefill 在信封布局下通过 gather/scatter 兜底（性能后置）。

关键文件：

- python/sglang/srt/mem_cache/layout/page_major.py（模块 缓存布局；类别 source；类型 core-logic；符号 _prod, mha_entry_bytes, build_page_major_mha_views, mamba_entry_bytes）：新增页面主序 MHA 和 Mamba 状态 strided view 构建器，定义布局几何，是 PR 的核心抽象。
- python/sglang/srt/mem_cache/memory_pool.py（模块 内存池；类别 source；类型 dependency-wiring；符号 _store_kv_layer, _move_kv_cache_impl, PageMajorMHATokenToKVPool, init）：集成新布局的入口：新增 PageMajorMHATokenToKVPool 子类，修改 MambaPool 初始化以支持信封分支，是布局与现有池架构的桥接点。
- python/sglang/srt/server_args.py（模块 配置管理；类别 source；类型 core-logic；符号 _handle_page_major_kv_layout）：添加 --enable-page-major-kv-layout 标志和验证器，强制后端依赖，是启用布局的配置入口。
- test/registered/unit/mem_cache/test_store_cache_4d.py（模块 存储缓存；类别 test；类型 test-coverage；符号 _legacy_advanced_indexing_write, TestStoreCache4D, _make_view_and_cache, _check_parity）：新增 store_cache_4d 内核的 byte-identical 测试，覆盖 page_size=1、>1、int32/int64 loc、bf16/fp8、不对称 dim 等场景。
- test/registered/unit/mem_cache/test_triton_kernel_layout.py（模块 内核布局；类别 test；类型 test-coverage；符号 TestTritonKernelLayoutParity, _setup_decode_inputs, _run_decode, test_decode_3d_vs_4d_ps1_byte_identical）：新增解码 / 扩展内核在 3-D、4-D ps=1、4-D ps>1 之间的 byte-identical 测试，确保 constexpr 折叠生效。
- test/registered/unit/mem_cache/test_page_major_layout.py（模块 页面布局；类别 test；类型 test-coverage；符号 _make_mha_views, TestPageMajorMHAViews, test_view_shapes, test_no_aliasing_ps1）：新增 CPU 端布局视图和移动操作的核验测试，确保 view shape、无别名、页内寻址正确。

关键符号：build_page_major_mha_views, build_page_major_mamba_views, mha_entry_bytes, mamba_entry_bytes, store_cache_4d, PageMajorMHATokenToKVPool._store_kv_layer, _handle_page_major_kv_layout, _extract_kv_strides

关键源码片段

python/sglang/srt/server_args.py

添加 --enable-page-major-kv-layout 标志和验证器，强制后端依赖，是启用布局的配置入口。

```
def _handle_page_major_kv_layout(self):
    """验证并强制执行 page-major 布局的后端依赖。

    当未启用时直接返回；启用时要求所有 attention 和 linear attention
    backend 均为 Triton，否则抛出断言。
    """
    if not self.enable_page_major_kv_layout:
```

```
        return

    # 收集 attention backend 并排除 None
    backends = {
        self.attention_backend,
        self.prefill_attention_backend,
        self.decode_attention_backend,
    }
    backends.discard(None)
    # page-major 的 strided 4-D view 仅被 Triton 内核支持
    assert backends <= {"triton"}, (
        f"--enable-page-major-kv-layout 要求 Triton attention backend，"
        f"当前为 {sorted(backends)}。请传入 --attention-backend triton。"
    )

    # 对 linear attention / Mamba backend 也要求 Triton
    linear_backends = {
        self.linear_attn_backend,
        self.linear_attn_decode_backend,
        self.linear_attn_prefill_backend,
        self.mamba_backend,
    }
    linear_backends.discard(None)
    assert linear_backends <= {"triton"}, (
        f"--enable-page-major-kv-layout 要求 Triton linear attention / "
        f"Mamba backend，当前为 {sorted(linear_backends)}。"
        f"请传入 --linear-attn-backend triton 和 --mamba-backend triton。"
    )
```

评论区精华

Reviewer ByronHsu 直接批准（approved），未提出评论。PR 作者自审并自行合并。

- 暂无高价值评论线程

风险与影响

- 风险：默认关闭时无回归风险（constexpr 折叠保证 SASS 一致）。启用后面临以下风险：必须使用 Triton backend，不符合时断言失败；不支持 FP4 KV 缓存和 speculative-decode 目标验证路径；GDN prefill 在信封布局下有额外 .contiguous() gather/scatter 开销（标记 TODO，计划后续做 stride-aware）；CPU offload 等路径（get_contiguous_buf_infos, get_cpu_copy, load_cpu_copy）抛出 NotImplementedError，依赖这些功能的代码需适配；新布局修改了 MambaPool 的初始化控制流，需确保启用时的路径覆盖所有混合模型（已通过 Qwen3.5 和 gpt-oss 端到端测试）。
- 影响：用户影响：默认无影响；启用后必须使用 --attention-backend triton --linear-attn-backend triton --mamba-backend triton，且无法与 FP4、spec-dec 目标验证路径同时使用。系统影响：新布局可能改善页面粒度操作的局部性，但 GDN 预 fill 初期有额外 copy。团队影响：内存池需要维护两套布局路径（通过子类而非内部分支），后续

编码器需意识到两种布局的存在。

- 风险标记：核心缓存布局变更，必须 Triton backend，GDN prefill 有额外 copy，CPU offload 抛出 NotImplementedError，暂不支持 FP4 和 speculative decode

关联脉络

- 暂无明显关联 PR