PR #38831 完整报告

vllm-project/vllm

[ModelRunnerV2][Hybrid model] Support kernel block size in hybrid model

合并时间：2026-05-28 08:55

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/38831

执行摘要

- 一句话：支持 ModelRunnerV2 混合模型的 kernel block size
- 推荐动作：值得精读。重点关注 init_attn_backend 的重构思路（分离 group 发现与 cg support）以及 BlockTables 中 kernel_block_sizes 的集成方式。设计决策（generator vs list、numpy vs list 回退）的权衡过程也值得借鉴。后续 PR 将基于此继续完善混合模型支持。

功能与动机

PR body 明确指出：This pr is a follow-up pr of #35520, resolving the discussion about kernel block size support. The motivation is to support kernel block size in hybrid model, which allows more attention backends to work with hybrid model. 此前 V2 model runner 完全禁止了 hybrid/mamba 模型，通过此 PR 可解禁大部分场景。

实现拆解

1. 重构 init_attn_backend（attn_utils.py）：将原来的发现 attention group + 创建 metadata builder + 确定 cudagraph support 合并函数拆解为 init_attn_backend（仅发现 group）和新函数 get_attention_cg_support_info（构建 metadata builder 并确定 cudagraph 支持）。这简化了 kernel_block_sizes 的准备流程。
2. 传递 kernel_block_sizes（block_table.py）：在 BlockTables 构造函数中新增 kernel_block_sizes 参数，计算每个 KV cache group 的 blocks_per_kv_block = block_size // kernel_block_size。在 append_block_ids 中展开 block IDs 时使用 kernel_block_sizes 映射，确保 attention kernel 能在正确的粒度上访问块。
3. 修正 gather 步长：_gather_block_tables_kernel 中不再单独传入 max_num_blocks，而是从 stride 获取，以兼容每组的 max_num_blocks 可能不同（因对齐规则和 MambaSpec 额外块）。
4. 配置层放宽限制（vllm/config/vllm.py）：将原来完全禁止 has_inner_state 模型的条件精确到仅当 mamba_cache_mode == "align"（即对齐前缀缓存模式）时才报不支持，从而允许混合模型在非 align 模式下使用 V2 运行器。
5. 适配调用方：同步更新 model_runner.py 和 speculator.py 中的调用，移除不再返回的 attn_backends，并使用新的三元素解包。init_kv_cache 等函数也改用 attn_groups 列表。

关键文件：

- vllm/v1/worker/gpu/attn_utils.py（模块 注意力后端；类别 source；类型 core-logic；符号 init_attn_backend, get_attention_cg_support_info, _reshape_kv_cache, init_kv_cache）：核心重构文件。重写 init_attn_backend，分离出 get_attention_cg_support_info，精简返回参数；init_kv_cache 改用 attn_groups。
- vllm/v1/worker/gpu/model_runner.py（模块 模型运行器；类别 source；类型 data-contract；符号 initialize_kv_cache）：调用 init_attn_backend 的入口，适配新返回值；同时传递 kernel_block_sizes 给 BlockTables。
- vllm/v1/worker/gpu/block_table.py（模块 块表；类别 source；类型 core-logic；符号 BlockTables.init, append_block_ids）：新增 kernel_block_sizes 参数，计算 blocks_per_kv_block，并用于展开 block IDs。
- vllm/config/vllm.py（模块 配置；类别 source；类型 configuration；符号 _get_v2_model_runner_unsupported_features）：修改 V2 模型运行器对 hybrid model 的限制条件，仅当 align cache mode 时报告不支持。
- vllm/v1/worker/gpu/spec_decode/eagle/speculator.py（模块 推测解码；类别 source；类型 dependency-wiring；符号 set_attn）：适配 init_attn_backend 返回值变化，移除对 attn_backends 的依赖。

关键符号：init_attn_backend, get_attention_cg_support_info, _reshape_kv_cache, init_kv_cache, BlockTables.init, BlockTables.append_block_ids, _gather_block_tables_kernel, _get_v2_model_runner_unsupported_features

关键源码片段

vllm/v1/worker/gpu/block_table.py

新增 kernel_block_sizes 参数，计算 blocks_per_kv_block，并用于展开 block IDs。

```
class BlockTables:
    def __init__(
        self,
        block_sizes: list[int],
        kernel_block_sizes: list[int], # new parameter
        max_num_reqs: int,
        max_num_batched_tokens: int,
        max_num_blocks_per_group: list[int],
        device: torch.device,
        cp_size: int = 1,
        cp_rank: int = 0,
        cp_interleave: int = 1,
    ):
        self.kernel_block_sizes = kernel_block_sizes
        # kernel_block_sizes length must match block_sizes
        assert len(kernel_block_sizes) == len(block_sizes)

        self.num_kv_cache_groups = len(block_sizes)
        # 计算每个 kv block 包含多少个 kernel block
        self.blocks_per_kv_block = [
```

```
            bs // kbs for bs, kbs in zip(block_sizes, kernel_block_sizes)
        ]

        # block table 的列数改为 max_num_blocks * blocks_per_kv_block
        for i in range(self.num_kv_cache_groups):
            max_num_blocks = (
                max_num_blocks_per_group[i] * self.blocks_per_kv_block[i]
            )
            block_table = StagedWriteTensor(
                (self.max_num_reqs, max_num_blocks),
                dtype=torch.int32,
                device=device,
            )
            self.block_tables.append(block_table)

    def append_block_ids(self, req_index, new_block_ids, overwrite=False):
        for i in range(self.num_kv_cache_groups):
            start = self.num_blocks.np[i, req_index] if not overwrite else 0
            block_ids = new_block_ids[i]
            bpk = self.blocks_per_kv_block[i]
            if bpk > 1:
                # 展开：每个 kv_block 对应 bpk 个 kernel block
                block_ids = [
                    b * bpk + k for b in block_ids for k in range(bpk)
                ]
            self.block_tables[i].stage_write(req_index, start, block_ids)
            self.num_blocks.np[i, req_index] = start + len(block_ids)
```

评论区精华

- generator 耗尽问题：ivanium 指出在 init_kv_cache 中 flattened_attn_groups = (group for groups in attn_groups for group in groups) 是 generator，在 _reshape_kv_cache 中迭代后会耗尽，导致后续 _update_hybrid_attention_layout 收到空迭代器。MengqingCao 确认后改用 list comprehension 避免此问题。
- 映射计算性能权衡：BlockTables 中 _map_to_kernel_blocks 改用 numpy 还是保持 list 引发讨论。MengqingCao 给出基准测试显示 numpy 在 block ≥ 32 时更快，但 njhill 优先关注简单性，最终决定保留列表推导式并承诺后续 PR 优化。
- Mamba align cache mode 处理：gemini-code-assist 指出 Mamba 的 block 分配未正确处理 align 模式，可能引起运行时内存不足。MengqingCao 回应前缀缓存支持将在后续 PR 完成，本 PR 专注于基础支持。
- 类型注解修正：ivanium 发现 group_map 的键类型应为 tuple[str, KVCacheSpec] 而非 tuple[tuple[str, str], KVCacheSpec]，已修复。
- 配置检查强化：ivanium 建议精准拦截 mamba_cache_mode == "align" 而不是完全禁用 hybrid model，该建议被采纳。
    - attn_groups generator 导致后续迭代为空 (correctness): MengqingCao 确认后改为 list comprehension，保证可多次迭代。

- 映射 block IDs 时使用 numpy 还是 list 的性能权衡 (performance): 最终采用简单 list 推导式，后续 PR 考虑 numpy 优化。
- Mamba block 分配未正确处理 align cache mode (correctness): 承认问题但推迟解决，本 PR 仅提供基础支持。
- group_map 类型注解错误 (style): MengqingCao 已修正。
- V2 model runner 是否应完全禁用 hybrid model (design): 仅当 mamba_cache_mode == "align" 时报不支持，其他 hybrid 模型可正常使用。

风险与影响

- 风险：
  - 回归风险（核心路径）：KV cache 初始化和 block table 计算路径被重写，虽然经过了 Qwen3-Next 和 Qwen3.5 的 gsm8k 精度验证，但未覆盖所有模型和 attention backend。
  - 性能风险：最终采用了 list 推导式展开 kernel block ID，在 batch size 较小时性能可接受；但大数据量场景下可能不如 numpy 高效，已计划后续优化。
  - 兼容性风险：init_attn_backend 返回值从 4 元组变为 3 元组，所有调用点已同步更新。但仍有部分功能（MTP、prefix cache align、disaggregated prefill）标记为 TODO，混合模型在这些场景下可能仍受限。
  - 缺少测试覆盖：没有新增针对 kernel_block_sizes 的单元测试，仅依赖现有 gsm8k 评估。
- 影响：
  - 用户：V2 model runner 现在可以运行 hybrid/mamba 模型（如 Qwen3-Next），并兼容更多 attention backend；但 align 模式下的 prefix caching 仍不可用。
  - 系统：BlockTables 现在依赖 kernel_block_sizes，增加了内存布局的灵活性，但也增加了初始化时计算 blocks_per_kv_block 的开销。
  - 团队：为后续完全支持 hybrid model 搭建了桥梁，明确的 TODO 列表指导后续工作。
  - 风险标记：核心路径变更，缺少测试覆盖，待完成项多

关联脉络

- PR #35520 previous PR introducing initial block size support: 本 PR 是 #35520 的后续，解决其 review 中遗留的 kernel block size 问题。
- PR #43084 fix CI issue referenced in comments: 作者提到 CI 失败已被 #43084 修复，本 PR 的 CI 依赖此修复。