PR #37898 完整报告

vllm-project/vllm

[Hybrid] Marconi-style admission policy for hybrid cache

合并时间：2026-06-11 01:03

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/37898

执行摘要

- 一句话：为混合缓存实现 Marconi 式共享前缀准入策略，提升 Qwen 等模型缓存命中率
- 推荐动作：值得精读。该 PR 展示了如何在不修改 kernel 的情况下利用现有缓存机制实现高级缓存准入策略，设计思路可推广到其他 hybrid attention 模型。

功能与动机

Marconi 论文 (https://arxiv.org/abs/2411.19379) 提出两种有效缓存准入策略：最后状态和共享前缀。vLLM 此前只支持最后状态缓存，缺少共享前缀缓存，导致系统提示等重复前缀场景无法充分利用前缀缓存。本 PR 实现共享前缀缓存，以进一步减少预填充时间。

实现拆解

1. 在 KVCacheCoordinator._get_cache_hit_blocks 中新增 longest_hit_length 记录所有 attention 组的最长缓存命中，计算 num_uncached_common_prefix_tokens 并暴露为属性。
2. 在 Scheduler.schedule 中，若模型包含 Mamba 层，从协调器读取 num_uncached_common_prefix_tokens。
3. 在 Scheduler._mamba_block_aligned_split 中接收该参数，若未缓存公共前缀 ≥ block_size 且当前调度长度更长，则截断调度长度以强制缓存该前缀，保持块对齐。
4. 新增测试 test_hybrid_cache_mamba_align_shared_prefix_detection，覆盖共享前缀检测及调度调整逻辑。

关键文件：

- vllm/v1/core/sched/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 _mamba_block_aligned_split, schedule）：核心调度器，修改了 _mamba_block_aligned_split 函数以支持共享前缀缓存的调度对齐
- vllm/v1/core/kv_cache_coordinator.py（模块 缓存协调器；类别 source；类型 core-logic；符号 _get_cache_hit_blocks, num_uncached_common_prefix_tokens）：缓存协调器，追踪最长缓存命中长度并暴露未缓存公共前缀长度，为调度器提供决策依据
- tests/v1/core/test_prefix_caching.py（模块 测试；类别 test；类型 test-coverage；符号 test_hybrid_cache_mamba_align_shared_prefix_detection）：新增 test_hybrid_cache_mamba_align_shared_prefix_detection 测试，验证共享前缀检测和调度调整

关键符号：_mamba_block_aligned_split, _get_cache_hit_blocks, test_hybrid_cache_mamba_align_shared_prefix_detection

关键源码片段

vllm/v1/core/kv_cache_coordinator.py

缓存协调器，追踪最长缓存命中长度并暴露未缓存公共前缀长度，为调度器提供决策依据

```
def _get_cache_hit_blocks(self, block_hashes, max_cache_hit_length):
    # ... 初始化变量 ...
    longest_hit_length = 0 # 新增：记录所有 attention 组中的最长命中
    while True:
        curr_hit_length = hit_length
        for idx, (spec, group_ids, manager_cls, use_eagle) in enumerate(
            self.attention_groups
        ):
            # ... 原有逻辑：查找每组最长缓存命中 ...
            hit_blocks = manager_cls.find_longest_cache_hit(...)
            _new_hit_length = len(hit_blocks[0]) * spec.block_size
            # ... 更新 curr_hit_length ...
            # 新增：更新全局最长命中
            longest_hit_length = max(longest_hit_length, curr_hit_length)
        # ... 循环终止条件 ...
    # ... 截断 Full Attention 块 ...
    # 新增：计算未缓存的公共前缀 token 数
    self.num_uncached_common_prefix_tokens = longest_hit_length - hit_length
    return (blocks_tuple, hit_length)
```

评论区精华

命名讨论

tdoublep 建议将参数名 mamba_tokens_lag 改为 num_uncached_common_prefix_tokens，语义更清晰，已采纳。

未初始化变量风险

gemini-code-assist 指出 mamba_tokens_lag 可能未定义导致 UnboundLocalError；作者说明该路径受条件保护，后续版本通过属性方式彻底规避。

变量复用可读性

gemini-code-assist 指出 num_new_local_computed_tokens 复用易混淆；作者重构为直接使用协调器属性，消除复用。

协调器返回值设计

tdoublep 和 yannicks1 讨论 get_computed_blocks 是否应返回两个长度；最终采用协调器属性方式，简化接口。

条件判断粒度

yannicks1 建议使用 need_mamba_block_aligned_split；作者认为 has_mamba_layers 更通用，以覆盖非 align 模式。

- 重命名参数 mamba_tokens_lag (design): 作者同意并修改。
- 未初始化变量风险 (correctness): 作者解释该代码块受条件保护，后续版本通过属性方式避免。
- 变量复用导致可读性差 (style): 作者重构为直接返回 num_uncached_common_prefix_tokens，移除了复用。
- 协调器返回值设计 (design): 作者改为在协调器上设置属性 num_uncached_common_prefix_tokens，调度器直接读取。
- 条件判断使用 has_mamba_layers vs need_mamba_block_aligned_split (design): 作者解释 has_mamba_layers 更通用，因为需要获取最长命中长度，即使 mamba 模式不是 align。
- 测试中断言顺序 (testing): 作者调整了测试中的断言顺序。

风险与影响

- 风险：
  1. 变量未初始化：虽当前路径受保护，但未来条件变更可能导致 UnboundLocalError，最终版本通过属性固化。
  2. 依赖 block_size 对齐：优化仅在 num_uncached_common_prefix_tokens >= block_size 时生效，小前缀无法获益，逻辑正确。
  3. 协调器类型假设：调度器通过 has_mamba_layers 间接判断协调器类型，若引入非 Mamba 混合模型需调整。
  4. 测试覆盖：新增测试覆盖主要场景，但边缘情况（如零前缀、非对齐边界）未充分测试。- 影响：用户：使用混合缓存的模型（如 Qwen）在启用 Prefix Caching 后预填充延迟降低 28%~40%。系统：仅在模型含 Mamba 层且缓存模式为 align 时触发，不影响其他路径。团队：需维护协调器属性及调度器逻辑，但代码量小，易于理解。
- 风险标记：变量未初始化风险，依赖 block_size 对齐，协调器类型假设

关联脉络

- 暂无明显关联 PR