PR #39562 完整报告

vllm-project/vllm

[Bugfix]: Fix assertion in MambaManager.allocate_slots()

合并时间：2026-06-08 12:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/39562

执行摘要

- 一句话：修复 MambaManager 因推测轮次动态变化导致的断言失败
- 推荐动作：该 PR 修复关键崩溃问题，修改简洁且测试覆盖充分，建议尽快合并。

功能与动机

Issue #39271 报告在使用 Qwen3.5-4B 启用后缀解码时触发 AssertionError。根因是调度器按当前轮次草稿令牌数计算 num_new_tokens，当草稿数下降时 num_tokens_main_model 减少，导致 num_required_blocks 小于已有块数，触发断言。

实现拆解

1. 修改核心断言：在 vllm/v1/core/single_type_kv_cache_manager.py 的 allocate_new_blocks() 方法中，将 if num_required_blocks == len(req_blocks) 改为 if num_required_blocks <= len(req_blocks)，并移除下方的 assert num_required_blocks > len(req_blocks) 语句。这样当所需块数不大于当前块数时直接返回空列表，避免崩溃。
2. 添加回归测试：在 tests/v1/core/test_prefix_caching.py 中新增 test_hybrid_model_mamba_align_with_dynamic_draft_tokens 函数，模拟预填充 → 高推测轮次（16 个草稿） → 低推测轮次（1 个草稿）场景，验证分配不崩溃且返回空块组。

关键文件：

- vllm/v1/core/single_type_kv_cache_manager.py（模块 缓存管理；类别 source；类型 core-logic；符号 allocate_new_blocks）：核心 bugfix 位置，修改 mamba_align 模式下块分配断言，将 == 改为 <= 以兼容动态草稿数。
- tests/v1/core/test_prefix_caching.py（模块 前缀缓存；类别 test；类型 test-coverage；符号 test_hybrid_model_mamba_align_with_dynamic_draft_tokens）：新增回归测试，覆盖动态草稿数场景，验证 fix 正确性。

关键符号：allocate_new_blocks, test_hybrid_model_mamba_align_with_dynamic_draft_tokens

关键源码片段

vllm/v1/core/single_type_kv_cache_manager.py

核心 bugfix 位置，修改 mamba_align 模式下块分配断言，将 == 改为 <= 以兼容动态草稿数。

```
def allocate_new_blocks(
    self, request_id: str, num_tokens: int, num_tokens_main_model: int
) -> list[KVCacheBlock]:
    assert isinstance(self.kv_cache_spec, MambaSpec)
    if self.mamba_cache_mode != "align":
        if self.num_speculative_blocks > 0:
            num_tokens += self.block_size * self.num_speculative_blocks
        return super().allocate_new_blocks(
            request_id, num_tokens, num_tokens_main_model
        )
    else:
        num_tokens = num_tokens_main_model
        req_blocks: list[KVCacheBlock] = self.req_to_blocks[request_id]
        # NOTE(tdouble): 这是一个高估，因为 num_tokens 包含可能被拒绝的草稿令牌
        num_required_blocks = (
            cdiv(num_tokens, self.block_size) + self.num_speculative_blocks
        )
        # 如果上一轮过分配，当前所需块数可能 <= 已有块数，此时返回 [] 不分配新块
        if num_required_blocks <= len(req_blocks):
            return []
        else:
            prev_block_len = len(req_blocks)
            blocks_allocated = request_id in self._allocated_block_reqs
            if blocks_allocated:
                self.last_state_block_idx[request_id] = (
                    prev_block_len - 1 - self.num_speculative_blocks
                )
            elif prev_block_len > 0:
                self.last_state_block_idx[request_id] = prev_block_len - 1
            num_skipped_blocks = (
                num_required_blocks - self.num_speculative_blocks - 1
            )
            if prev_block_len < num_skipped_blocks:
                req_blocks.extend(
                    [self._null_block for _ in range(prev_block_len, num_skipped_blocks)]
                )
            # ... 后续复用 speculative blocks 的逻辑保持不变
```

tests/v1/core/test_prefix_caching.py

新增回归测试，覆盖动态草稿数场景，验证 fix 正确性。

```
def test_hybrid_model_mamba_align_with_dynamic_draft_tokens():
    """Regression test for https://github.com/vllm-project/vllm/issues/39271.

    当后缀解码的草稿令牌数动态变化时，MambaManager 不应崩溃。
    模拟 prefill → 16 个草稿 → 1 个草稿 的序列。
    """
    block_size = 16
    num_blocks = 30
```

```
kv_cache_config = _make_hybrid_kv_cache_config(
    block_size, num_blocks, ["full", "mamba_align"]
)
manager = KVCacheManager(
    kv_cache_config,
    max_model_len=8192,
    enable_caching=True,
    hash_block_size=block_size,
    scheduler_block_size=block_size,
)
hash_fn = sha256
all_token_ids = [i for i in range(3) for _ in range(block_size)] + [3] * 7
req0 = make_request("0", all_token_ids, block_size, hash_fn)
computed_blocks, num_computed_tokens = manager.get_computed_blocks(req0)
assert num_computed_tokens == 0
blocks = manager.allocate_slots(
    req0, len(all_token_ids), num_computed_tokens, computed_blocks
)
assert blocks is not None

# prefill forward finished
req0.append_output_token_ids([1])
req0.num_computed_tokens = len(all_token_ids)

# Round1: 提出 16 个草稿，只接受一个
req0.spec_token_ids = [4] * 16
blocks = manager.allocate_slots(req0, num_new_tokens=16, num_new_computed_tokens=0)
assert blocks is not None
req0.append_output_token_ids([4])
req0.num_computed_tokens += 1

# Round2: 只提出 1 个草稿，此时所需块数可能少于已有块，不应崩溃
req0.spec_token_ids = [5] * 1
blocks = manager.allocate_slots(req0, num_new_tokens=1, num_new_computed_tokens=0)
# 验证不分配新块：返回空块组
assert blocks is not None and all(len(group) == 0 for group in blocks.blocks)

manager.free(req0)
```

评论区精华

- ivanium（审核人）指出 let's remove this obsolete assertion，确认删除断言。
- gemini-code-assist[bot]建议在测试中增加 `assert all(len(group) == 0 for group in blocks.blocks)` 以验证无新块分配，该建议被采纳。
- 移除过时断言 (correctness): njhill 在后续提交中移除了断言。
- 加强测试断言 (testing): 该建议被采纳，最终测试包含此断言。

风险与影响

- 风险：修改仅涉及 MambaManager.allocate_new_blocks 中的条件判断，与父类行为一致，回归风险低。但需注意该逻辑与推测解码调度器耦合，若未来其他推测方法引入不同草稿数变化模式需额外验证。
- 影响：用户侧：修复了使用后缀解码的混合模型（如 Qwen3.5）的崩溃问题，增强稳定性。系统侧：无明显性能影响，过分配块保留后重用。
- 风险标记：核心路径变更，动态推测解码兼容性

关联脉络

- 暂无明显关联 PR