PR #47728 完整报告

vllm-project/vllm

[Bugfix][V1] Free out-of-window blocks on the processed-token basis under async scheduling

合并时间：2026-07-08 20:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47728

执行摘要

- 一句话：修复异步调度下窗口外 KV 块提前释放的 bug
- 推荐动作：该 PR 是一个关键的正确性修复，建议团队根据长序列场景的性能回归考虑优化 admission 逻辑（如条件性宽松上线），并鼓励在此基础上进一步清理 MambaManager 中的局部修补代码。代码审查中 njhill 的简化 commit 值得学习，可作为跨模块参数传递设计的参考。

功能与动机

PR body 指出：'With async scheduling (now default) or PP, the scheduler runs ahead of the GPU: request.num_computed_tokens is advanced at schedule time. remove_skipped_blocks (SWA / chunked-local / mamba) used this optimistic count as the free boundary, causing two bugs: load-WAR 和 spec-rejection misfree'。同时关联 Issue #47282（load-WAR）和 #50（原始修复）。核心动机是消除调度乐观计数导致的 KV 块提前回收，避免数据竞争和静默错误。

实现拆解

1. 在 Request 中新增 num_in_flight_tokens 字段：在 vllm/v1/request.py 中添加属性，并在 vllm/config/vllm.py 中新增 max_in_flight_tokens 配置项，用于串联调度器与缓存管理器间的 token 积压视图。
2. 在调度器中维护 in-flight 统计：在 Scheduler._update_after_schedule 中递增 num_in_flight_tokens，在 update_from_output 中递减；同时在 _connector_finished 中使用 max(0, num_computed_tokens - num_in_flight_tokens) 作为 remove_skipped_blocks 的已处理边界，确保释放时已考虑未完成步骤。
3. 在缓存管理器中统一入参并修改释放逻辑：将 remove_skipped_blocks 的参数 total_computed_tokens 重命名为 processed_computed_tokens，明确语义；在 KVCacheManager.allocate_slots 中调用时传入 max(0, total_computed_tokens - request.num_in_flight_tokens)，从而实现“已处理 token 基础”释放。同时移除 MambaManager.remove_skipped_blocks 中关于 spec tokens 的局部减法（已被新机制涵盖）。
4. 收紧 admission 上限：为 SWA 和 chunked-local 在初始化时计算考虑最大并发批次的 admission 上限，即 W - 1 + max_concurrent_batches * max_num_batched_tokens，避免在大量重叠批次时过度放行请求。

5. 新增并更新测试覆盖：新增tests/v1/core/test_swa_inflight_window_free.py（7 个测试），覆盖 SWA/chunked-local 的 in-flight 释放等待、admission 上限以及 connector-finish 场景；修改 tests/v1/e2e/general/test_mamba_prefix_cache.py，将原有 golden 拆分为 sync/async 两个变体，避免单一 golden 无法匹配两种模式。

关键文件：

- vllm/v1/core/kv_cache_manager.py（模块 缓存层；类别 source；类型 core-logic；符号 allocate_slots, remove_skipped_blocks, KVCacheManager.init）：核心修改：在 allocate_slots 中使用 settled basis 调用 remove_skipped_blocks，参数重命名以明确语义。
- vllm/v1/core/sched/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 _update_after_schedule, update_from_output, _connector_finished, Scheduler.init）：核心修改：维护 num_in_flight_tokens 的加减，并在 _connector_finished 中使用 settled basis。
- tests/v1/core/test_swa_inflight_window_free.py（模块 测试；类别 test；类型 test-coverage；符号 _make_model_runner_output, _create_swa_scheduler, _create_chunked_scheduler, _num_null_blocks）：新增完整测试套件，覆盖 SWA/chunked-local 的 in-flight 释放等待、admission 上限和 connector-finish 场景，是验证修复的关键。
- vllm/v1/request.py（模块 请求模型；类别 source；类型 core-logic；符号 num_in_flight_tokens）：新增 num_in_flight_tokens 字段，作为修复核心数据结构。
- vllm/v1/core/single_type_kv_cache_manager.py（模块 缓存层；类别 source；类型 core-logic；符号 remove_skipped_blocks, RSWAManager.remove_skipped_blocks, MambaManager.remove_skipped_blocks）：参数重命名和清理 MambaManager 中过时的 spec token 减法逻辑，体现统一后的语义。

关键符号：Request.num_in_flight_tokens, Scheduler._update_after_schedule, Scheduler.update_from_output, Scheduler._connector_finished, KVCacheManager.allocate_slots, KVCacheManager.remove_skipped_blocks, SingleTypeKVCacheManager.remove_skipped_blocks, RSWAManager.remove_skipped_blocks, MambaManager.remove_skipped_blocks, test_num_in_flight_tokens_accounting, test_swa_free_waits_for_in_flight_step

关键源码片段

vllm/v1/core/kv_cache_manager.py

核心修改：在 allocate_slots 中使用 settled basis 调用 remove_skipped_blocks，参数重命名以明确语义。

```
# From vllm/v1/core/kv_cache_manager.py
class KVCacheManager:
    def allocate_slots(...):
        # ...
        # Free on the processed-token basis: in-flight steps' attention windows
        # still read blocks below the optimistic boundary, and rejected spec
```

```
        # tokens can roll it back.
        self.coordinator.remove_skipped_blocks(
            request.request_id,
            max(0, total_computed_tokens - request.num_in_flight_tokens),
            num_prompt_tokens=request.num_prompt_tokens,
        )
        # ...
```

tests/v1/core/test_swa_inflight_window_free.py

新增完整测试套件，覆盖 SWA/chunked-local 的 in-flight 释放等待、admission 上限和 connector-finish 场景，是验证修复的关键。

```
# tests/v1/core/test_swa_inflight_window_free.py
# Validate that in-flight tokens are correctly tracked

NUM_PROMPT_TOKENS = 100
BLOCK_SIZE = 16
SLIDING_WINDOW = 16


def test_num_in_flight_tokens_accounting():
    scheduler = create_scheduler(async_scheduling=True)
    request = create_requests(num_requests=1, num_tokens=NUM_PROMPT_TOKENS)[0]
    scheduler.add_request(request)

    out0 = scheduler.schedule() # prefill
    # After prefill schedule, all tokens are in-flight (not yet processed)
    assert request.num_in_flight_tokens == NUM_PROMPT_TOKENS
    # Async: decode scheduled before prefill output is processed
    out1 = scheduler.schedule()
    # One more token scheduled, still unprocessed
    assert request.num_in_flight_tokens == NUM_PROMPT_TOKENS + 1

    # Process the prefill output
    scheduler.update_from_output(out0, _make_model_runner_output(out0))
    # Now only the decode token is in-flight
    assert request.num_in_flight_tokens == 1
    scheduler.update_from_output(out1, _make_model_runner_output(out1))
    # All processed
    assert request.num_in_flight_tokens == 0
```

评论区精华

- 参数过度透传：ZJY0516 指出 max_concurrent_batches 仅在 admission 上限计算中需要，却被层层透传，中间层语义并不依赖它；njhill 赞同并提交简化 commit（00c404d）将其替换为统一 max_in_flight_tokens，并重命名 remove_skipped_blocks 的参数为 processed_computed_tokens。

- MambaManager 注释清理：njhill 建议移除 MambaManager 中针对 spec tokens 的局部减法注释，因为该逻辑已被新机制替代。
- 测试失败定位：合并后发现 test_mamba_prefix_cache 因 golden 不匹配失败，Saddss 确认该测试依赖释放时机，在 async 模式下行为已变，需将 golden 拆分为 async/sync 两个版本。
  - max_concurrent_batches 参数透传简化 (design): njhill 提交了简化 commit（00c404d），将 max_concurrent_batches 替换为 max_in_flight_tokens，并重命名 remove_skipped_blocks 参数为 processed_computed_tokens。
  - MambaManager 中 spec token 减法注释清理 (style): njhill 在简化 commit 中移除了相关注释。
  - 测试失败：test_mamba_prefix_cache 因 golden 不匹配失败 (testing): Saddss 提交了修复 commit，将测试拆分为 sync/async 变体，并更新了 golden 值。

风险与影响

- 风险：该 PR 修改了核心调度与释放路径，主要风险包括：
  1. 异步调度下性能退化：PR body 报告在接近最大长度请求时吞吐降低约 34%，源于更保守的 admission 上限；
  2. 同步调度回归：更改理论上不影响同步调度（in-flight 为 0），但需确保配置传递正确；
  3. 与未来 KV offload 方案冲突：_connector_finished 中使用 settled basis 可能与某些 KV 卸载实现产生时间差；
  4. 测试覆盖不全：虽然新增了单元测试，但真正的 e2e 长序列场景并未自动化，需要人工验证。 - 影响：影响范围涉及所有启用 SWA、Chunked-local、Mamba 等滑动窗口或局部注意力的模型，在异步调度下（默认启用）运行时行为改变。对于普通长度请求，性能无影响；对于长序列（接近最大模型长度），存在吞吐下降的 trade-off。团队需要更新 admission 上限的默认值设定（新增 max_in_flight_tokens 配置），但内部实现了向后兼容 fallback。整体影响中等偏大，但已被充分测试。 - 风险标记：核心调度释放路径变更，长序列吞吐回归，参数重命名破坏外部调用，与未来 KV 卸载兼容

关联脉络

- PR #45357 Request-level finish/preempt frees fence: Request-level finish/preempt frees 仍保留在 #45357 的 fence 保护下，未被此 PR 修改。
- PR #47653 BlockPool fence alternative: 作为 alternatives considered 中的一种方案（BlockPool fence），被本 PR 拒绝，用于对比设计选择。