# PR #43494 完整报告

vllm-project/vllm

[KV Connector] Keep MooncakeStore full hits block-aligned

合并时间：2026-05-24 14:15

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43494

## 执行摘要

- 一句话：修复 MooncakeStore 全命中时块对齐问题
- 推荐动作：对于使用 MooncakeStore 的团队，此 PR 值得精读以理解块对齐的设计取舍和配置清理策略；对于其他开发者，可作为 KV Connector 调度器维护的参考案例。

## 功能与动机

根因：当 MooncakeStore 命中整个 prompt 时，调度器将外部命中数从 num_tokens 减 1，使得 kvpool_cached_tokens 非块对齐。接收端基于块构建加载掩码，而 process_tokens() 可能产生部分尾部，导致不匹配和潜在错误。此修复通过将完整命中向下舍入到 (num_tokens - 1) // block_size * block_size 来保持块对齐，避免边界问题。

## 实现拆解

1. 调度器核心逻辑调整（scheduler.py）：移除了 _discard_partial_chunks 配置及其所有条件分支，在 get_num_new_matched_tokens 中当外部完全命中时，不再简单减 1，而是计算 (request.num_tokens - 1) // self._block_size * self._block_size 作为新命中值，确保块对齐；同时移除了 build_connector_meta 中基于该配置的补偿 +1 逻辑和条件计算。
2. Worker 补偿移除（worker.py）：在 get_finished 中删除了对 kvpool_cached_tokens 的非对齐补偿（之前当 token_len 为 kvpool_cached_tokens-1 时加回），现在直接使用 kvpool_cached_tokens 作为 token_len。
3. 回归测试新增（test_mooncake_store_scheduler.py）：添加了 _StubLookupClient 辅助类模拟外部查找，并新增两个测试用例验证全命中时块对齐（need_to_allocate 正确且 kvpool_cached_tokens 对齐）以及全命中 + 本地完全命中时跳过加载（need_to_allocate=0 且无 load_spec）。
4. 文档更新（mooncake_store_connector_usage.md）：删除了已移除的 discard_partial_chunks 配置说明。

关键文件：

- vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 get_num_new_matched_tokens, build_connector_meta, init）：核心变更：修改全命中时块对齐计算，移除 _discard_partial_chunks 配置及其所有相关分支，简化逻辑

- tests/v1/kv_connector/unit/test_mooncake_store_scheduler.py（模块 测试；类别 test；类型 test-coverage；符号 _StubLookupClient, init, lookup, test_full_external_hit_keeps_kvpool_cached_tokens_block_aligned）：新增两个回归测试，覆盖全命中时块对齐和零负载场景，防止回归
- vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/worker.py（模块 加载器；类别 source；类型 core-logic；符号 get_finished）：移除补偿逻辑，直接使用 kvpool_cached_tokens，与调度器保持一致
- docs/features/mooncake_store_connector_usage.md（模块 文档；类别 docs；类型 documentation）：删除已移除配置的文档说明，保持文档与代码一致

关键符号：get_num_new_matched_tokens, build_connector_meta, get_finished, test_full_external_hit_keeps_kvpool_cached_tokens_block_aligned, test_full_external_hit_with_full_local_hit_skips_load

## 关键源码片段

### vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/scheduler.py

核心变更：修改全命中时块对齐计算，移除 _discard_partial_chunks 配置及其所有相关分支，简化逻辑

```
# vllm/distributed/kv_transfer/kv_connector/v1/mooncake/store/scheduler.py
# 修改后的 get_num_new_matched_tokens 方法核心部分

def get_num_new_matched_tokens(
    self,
    request: Request,
    num_computed_tokens: int,
) -> tuple[int, bool]:
    """Check for external KV cache hit."""
    # 始终将 token_len 对齐到块边界，不再依赖 discard_partial_chunks 配置
    token_len = request.num_tokens // self._block_size * self._block_size
    if token_len < self._block_size:
        return 0, False

    num_external_hit_tokens = self.client.lookup(token_len, request.block_hashes)

    if num_external_hit_tokens == request.num_tokens:
        # 全命中时：保留一个子块尾部用于采样，并确保块对齐
        # 之前是简单的 `-= 1`，现在使用块边界舍入
        num_external_hit_tokens = max(
            0,
            (request.num_tokens - 1) // self._block_size * self._block_size,
        )

    if num_external_hit_tokens < num_computed_tokens:
        need_to_allocate = 0
    else:
```

```
        need_to_allocate = num_external_hit_tokens - num_computed_tokens

    # ... 日志和 load_spec 设置保持不变
    if need_to_allocate <= 0:
        return 0, False
    self.load_specs[request.request_id] = LoadSpec(
        vllm_cached_tokens=num_computed_tokens,
        kvpool_cached_tokens=num_external_hit_tokens, # 现在始终块对齐
        can_load=False,
    )
    return need_to_allocate, self.load_async
```

## tests/v1/kv_connector/unit/test_mooncake_store_scheduler.py

新增两个回归测试，覆盖全命中时块对齐和零负载场景，防止回归

```
# tests/v1/kv_connector/unit/test_mooncake_store_scheduler.py
# 新增测试：验证全命中时 kvpool_cached_tokens 块对齐

def test_full_external_hit_keeps_kvpool_cached_tokens_block_aligned():
    # 当外部存储完全命中整个 prompt 时，
    # 调度器必须保留一个 token 用于采样但保持块边界对齐。
    # 否则接收侧加载掩码会将 token_len 向下取整为
    # (num_tokens-1)//block_size，尾部块被丢弃，
    # 如果本地缓存覆盖了对齐前缀，会导致 key_list 为空
    #  （recv 线程中出现 ZeroDivisionError）。
    scheduler = _make_bare_scheduler()
    scheduler.load_async = True
    # 模拟 48 tokens 的完整命中
    scheduler.client = _StubLookupClient(hit_tokens=48)

    request = SimpleNamespace(
        request_id="req-0",
        num_tokens=48,
        block_hashes=[b"h0", b"h1", b"h2"],
    )

    need_to_allocate, load_async = scheduler.get_num_new_matched_tokens(
        request, num_computed_tokens=16
    )

    # 47 // 16 * 16 == 32 tokens 保留在外部存储中（保留子块尾部用于采样）
    # 32 - 16（本地已计算） == 16 需要加载
    assert need_to_allocate == 16
    assert load_async is True
    load_spec = scheduler.load_specs["req-0"]
    assert load_spec.vllm_cached_tokens == 16
    assert load_spec.kvpool_cached_tokens == 32
    # 关键断言：kvpool_cached_tokens 必须块对齐（可被 16 整除）
    assert load_spec.kvpool_cached_tokens % 16 == 0
```

## 评论区精华

Reviewer ivanium 提出了两点关键讨论：

- 针对 scheduler.py 中补偿逻辑的移除，评论“We should fix this logic by getting the block back rather than removing it directly?”，并在 worker.py 中同样质疑“same here. Should we fix the logic?”。
- 针对 __init__ 中移除 _discard_partial_chunks，建议“Can add an assertion in __init__ saying that we only support discard_partial_chunk”。 作者 Dao007forever 回应“Updated to remove discard_partial_chunk, and only support True.”，最终移除配置而非添加断言。ivanium 最终批准，但表示“It's a pity that we have to remove discard_partial_chunks for now, but we'd keep that in mind to add back in the future.”
- 全命中后补偿逻辑的移除 vs 保留并修复 (design): 作者选择移除补偿逻辑，并移除 discard_partial_chunks 配置，使块对齐行为统一。ivanium 最终批准但表示遗憾，计划未来可能加回。
- 是否在 __init__中添加断言确保 discard_partial_chunks 为 True (design): 作者回应已移除 discard_partial_chunk 配置并只支持 True，因此无需添加断言。该配置被彻底移除。

## 风险与影响

- 风险：主要风险在于移除了 discard_partial_chunks 配置选项，该配置之前默认 True 且被推荐使用，但如果有用户显式设为 False，升级后行为会强制变为 True（但一直推荐 True，影响极小）。此外，修改了全命中时的计算方式，可能改变边缘情况（如 num_tokens 刚好为 block_size 的倍数且完全命中时，之前减 1 会进入 0 或负数，现在使用 max(0, ...) 避免负值；但测试已覆盖该场景）。整体风险较低，且新增回归测试覆盖了主要路径。
- 影响：影响范围限定在使用 MooncakeStoreConnector 并启用 v1 KV 调度的用户。修复了因块不对齐可能导致的 ZeroDivisionError 或静默数据丢失，提升了可靠性。配置 discard_partial_chunks 不再可用，但该配置本应始终为 True，因此对用户无负面影响。代码量精简（-30 行），可维护性提升。
- 风险标记：核心调度逻辑变更，配置项移除

## 关联脉络

- PR #43392 [Mooncake] Add metrics for MooncakeStoreConnector operations: 同为 MooncakeStore 功能线，修改了 worker.py 等文件，与本 PR 有文件重叠
- PR #43433 Keep scheduler alive for delayed KV connector frees: 同为 kv-connector 调度器问题修复，涉及调度器状态管理，与本 PR 的调度器修改形成上下游关联