PR #51843 完整报告

vllm-project/vllm

[Bugfix] Disable fine-grained prefix-cache hits for incompatible hybrid KV layouts

合并时间：2026-08-12 19:08

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51843

执行摘要

- 一句话：混合 KV 布局禁用细粒度前缀命中，修复断言崩溃
- 推荐动作：值得精读。该 PR 是理解 vLLM v1 prefix cache 命中粒度体系（hash_block_size 与 scheduler_block_size 两级对齐）的很好入口，展示了一个 " 能力协商 + 统一开关降级 + 日志 + 回归测试 " 的完整 bugfix 模式。建议关注三个联动点：KVCacheCoordinator.enable_partial_hash_hits 的判定、_cache_hit_alignment_tokens 的对齐出口、Scheduler.mamba_partial_cache_hit 的调度守卫；后续若 SWA partial block hit 原型恢复，本 PR 的开关与日志将成为平滑升级的基础。

功能与动机

PR body 明确指出问题："Fine-grained prefix-cache hits are enabled for hybrid models containing Mamba align groups. However other groups, such as a sliding-window DSpark drafter, may use KV cache managers that only support block-aligned lookups. This previously caused an assertion when prefix caching was enabled." 即细粒度命中按 hash block 边界返回长度，而 sliding-window drafter 的 KV cache manager 只能按更大的 block 对齐查找，两者冲突导致断言崩溃。修复目标是让混合 KV 布局在保证正确性的前提下尽可能保留 prefix-cache 能力。

实现拆解

1. 能力判定重构（vllm/v1/core/kv_cache_coordinator.py）：原判定只看 " 是否有 Mamba align 组 + 无 context parallelism"；新逻辑先算出 has_partial_mamba_group，再枚举 self.single_type_managers，把 supports_fine_grained_hash_lookup 为 False 且 block_size 不等于 hash_block_size 的 manager 收进 unsupported_partial_hit_managers 集合。只要集合非空，enable_partial_hash_hits 置为 False，并用 logger.warning_once 告警一次。检测与告警都放在 enable_partial_hash_hits 为真的分支内，避免 dcp_world_size > 1 等场景误告警。
2. 默认值上提（vllm/v1/core/kv_cache_coordinator.py）：enable_partial_hash_hits 改为 KVCacheCoordinator 基类类属性，默认 False，保证任何未显式启用的路径都走最保守的 block 对齐行为，scheduler 侧不再需要 getattr 兜底。
3. 调度器联动（vllm/v1/core/sched/scheduler.py）：Scheduler.__init__ 里 mamba_partial_cache_hit 追加 coordinator.enable_partial_hash_hits 条件。该标志控制 scheduler 是否在 prompt 末尾的 hash 边界处增加 partial-tail 停顿点；降级后不再创建

这类记录，避免无法复用的缓存垃圾。

4. 对齐粒度出口（vllm/v1/core/kv_cache_coordinator.py）：_cache_hit_alignment_tokens property 依据 enable_partial_hash_hits 返回 hash_block_size 或 scheduler_block_size，保证所有 KV cache 组在查找时使用一致的命中对齐粒度。
5. 测试配套（tests/v1/core/prefix_cache/test_partial_prefix_cache_hits.py）：新增 test_hybrid_sliding_window_group_disables_partial_hash_hits，构造 full attention + Mamba align + sliding-window eagle drafter 三组 KV 布局，覆盖 " 禁用命中、按 mamba block 对齐回退、block 数严格对齐 " 三条断言；PR body 报告 23 个 partial 测试与 89 个 prefix 测试全部通过。

关键文件：

- vllm/v1/core/kv_cache_coordinator.py（模块 缓存协调；类别 source；类型 core-logic；符号 KVCacheCoordinator, enable_partial_hash_hits, _cache_hit_alignment_tokens）：修复的核心：新增全组 KV cache manager 能力校验，决定是否启用细粒度 prefix-cache 命中；同时把 enable_partial_hash_hits 提升为基类默认 False，并新增降级告警日志。
- vllm/v1/core/sched/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 Scheduler, mamba_partial_cache_hit）：一行关键守卫：mamba_partial_cache_hit 增加 coordinator.enable_partial_hash_hits 条件，确保降级后 scheduler 不再创建无法复用的 partial-tail 缓存记录。
- tests/v1/core/prefix_cache/test_partial_prefix_cache_hits.py（模块 前缀缓存；类别 test；类型 test-coverage；符号 test_hybrid_sliding_window_group_disables_partial_hash_hits）：新增 test_hybrid_sliding_window_group_disables_partial_hash_hits，端到端验证 full attention + Mamba align + sliding-window drafter 三组混合布局下的降级行为与 block 对齐命中。

关键符号：KVCacheCoordinator.init, KVCacheCoordinator._cache_hit_alignment_tokens, Scheduler.init, test_hybrid_sliding_window_group_disables_partial_hash_hits

关键源码片段

vllm/v1/core/kv_cache_coordinator.py

修复的核心：新增全组 KV cache manager 能力校验，决定是否启用细粒度 prefix-cache 命中；同时把 enable_partial_hash_hits 提升为基类默认 False，并新增降级告警日志。

```
# vllm/v1/core/kv_cache_coordinator.py 中的核心判定逻辑（head 版本整理）
class KVCacheCoordinator(ABC):
    """协调不同 KV cache group 的缓存布局、命中查找与分配。"""

    # 基类默认关闭细粒度前缀命中，子类按需打开；False 也是最保守的默认，
    # 保证未显式启用的路径一律走 block 对齐查找。
    enable_partial_hash_hits = False

    def __init__(self, kv_cache_config, max_model_len, ...):
        # 细粒度哈希命中需要同时满足三个前提：存在 Mamba "align" 组且其
```

```
        # block_size 大于 hash_block_size；未启用 context parallelism
        #  （dcp_world_size == 1）；每个 KV cache manager 都支持细粒度
        # 查找，否则 sliding-window drafter 这类组只能做 block 对齐查找，
        # 命中长度对不上会触发断言。
        has_partial_mamba_group = any(
            isinstance(g.kv_cache_spec, MambaSpec)
            and g.kv_cache_spec.mamba_cache_mode == "align"
            and g.kv_cache_spec.block_size > hash_block_size
            for g in kv_cache_config.kv_cache_groups
        )
        self.enable_partial_hash_hits = (
            dcp_world_size == 1 and has_partial_mamba_group
        )
        if self.enable_partial_hash_hits:
            # 收集所有要求 block 对齐查找的 manager。当 manager 的
            # block_size 大于 hash_block_size 时，hash 级细粒度查找无法
            # 满足，一旦存在就整体降级以保持各组命中长度一致。
            unsupported_partial_hit_managers = {
                type(manager).__name__
                for manager in self.single_type_managers
                if not manager.supports_fine_grained_hash_lookup
                and manager.block_size != hash_block_size
            }
            if unsupported_partial_hit_managers:
                self.enable_partial_hash_hits = False
                logger.warning_once( # 降级原因只告警一次，避免刷屏
                    "Disabling fine-grained prefix-cache hits because these KV "
                    "cache managers require block-aligned lookups: %s.",
                    ", ".join(sorted(unsupported_partial_hit_managers)),
                )

    @property
    def _cache_hit_alignment_tokens(self) -> int:
        # 细粒度命中时允许返回 hash 块对齐的长度；
        # 降级后必须保持 scheduler 块对齐，才能兼容所有组。
        return (
            self.hash_block_size
            if self.enable_partial_hash_hits
            else self.scheduler_block_size
        )
```

vllm/v1/core/sched/scheduler.py

一行关键守卫：mamba_partial_cache_hit 增加 coordinator.enable_partial_hash_hits 条件，确保降级后 scheduler 不再创建无法复用的 partial-tail 缓存记录。

```
# vllm/v1/core/sched/scheduler.py 中 Scheduler.__init__ 的联动守卫
# 仅当 coordinator 实际启用了细粒度命中时，才允许 scheduler 在 prompt
# 末尾的 hash 边界处追加 partial-tail 停顿点；一旦降级为 block 对齐
# 命中，partial 尾部记录无法被后续请求复用，反而浪费调度轮次。
```

```
self.mamba_partial_cache_hit = (
    self.need_mamba_block_aligned_split
    and self.hash_block_size < self.block_size
    and self.kv_cache_manager.coordinator.enable_partial_hash_hits
)
```

tests/v1/core/prefix_cache/test_partial_prefix_cache_hits.py

新增 test_hybrid_sliding_window_group_disables_partial_hash_hits，端到端验证 full attention + Mamba align + sliding-window drafter 三组混合布局下的降级行为与 block 对齐命中。

```
# tests/v1/core/prefix_cache/test_partial_prefix_cache_hits.py
def test_hybrid_sliding_window_group_disables_partial_hash_hits():
    # 构造 full attention、Mamba "align" 和 sliding-window drafter
    # 三组混合 KV 布局：drafter 的 block size 是 hash 块的 2 倍，只能
    # 做 block 对齐查找，因此细粒度命中必须被整体禁用。
    hash_block_size = 2
    sliding_window_block_size = 2 * hash_block_size
    mamba_block_size = 2 * sliding_window_block_size
    kv_cache_config = KVCacheConfig(
        num_blocks=64,
        kv_cache_tensors=[],
        kv_cache_groups=[
            KVCacheGroupSpec(
                ["full"],
                FullAttentionSpec(
                    block_size=hash_block_size, num_kv_heads=1,
                    head_size=1, dtype=torch.float32)),
            KVCacheGroupSpec(
                ["mamba"],
                MambaSpec(
                    block_size=mamba_block_size, shapes=(1, 1),
                    dtypes=(torch.float32,), mamba_cache_mode="align")),
            KVCacheGroupSpec(
                ["swa_draft"],
                SlidingWindowSpec(
                    block_size=sliding_window_block_size, num_kv_heads=1,
                    head_size=1, dtype=torch.float32,
                    sliding_window=sliding_window_block_size),
                is_eagle_group=True),
        ],
    )
    manager = make_kv_cache_manager(
        kv_cache_config=kv_cache_config, max_model_len=8192,
        enable_caching=True, hash_block_size=hash_block_size,
        use_eagle=True)

    # 先用首个请求填充缓存：第一段按 mamba_block_size 分配，
    # 后续段按剩余 token 补齐，之后释放。
```

```
tokens = list(range(3 * sliding_window_block_size))
request = make_request("0", tokens, hash_block_size, sha256)
computed_blocks, num_computed, _ = manager.get_computed_blocks(request)
assert not manager.coordinator.enable_partial_hash_hits # 关键断言：已降级
assert (manager.allocate_slots(
    request, mamba_block_size, num_computed, computed_blocks) is not None)
request.num_computed_tokens = mamba_block_size
manager.new_step_starts()
assert manager.allocate_slots(
    request, len(tokens) - mamba_block_size) is not None
request.num_computed_tokens = len(tokens)
manager.free(request)
manager.new_step_starts()

# 复用相同前缀的更长请求：命中长度必须等于 mamba_block_size，
# 且以 hash 块折算的 block 数严格对齐，不产生 partial-tail 记录。
cached_request = make_request(
    "1", tokens + [len(tokens), len(tokens) + 1], hash_block_size, sha256)
computed_blocks, num_computed, _ = manager.get_computed_blocks(cached_request)
assert num_computed == mamba_block_size
assert len(computed_blocks.blocks[0]) * hash_block_size == num_computed
```

评论区精华

njhill 审阅后总体认可（"Thanks @mgoin lgtm just minor comments"），随后 APPROVED。两处高质量建议：其一，针对 scheduler 中 getattr(coordinator, "enable_partial_hash_hits", True) 的写法，建议把默认值直接定义在 KVCacheCoordinator 基类上（False 是更合理默认），mgoin 回复 "Yup I agree"；其二，指出 unsupported_partial_hit_managers 集合在 dcp > 1 场景也会被构建、可能误告警，建议把构造与告警都收进 if 分支，mgoin 认可（"Nice catch!"）。此外 ivanium 在评论中提到团队已有 SWA partial block hit 原型但未合并，提示未来可以把细粒度命中恢复到 sliding-window 组上，否则当前降级逻辑让代码看起来更复杂。

- 基类默认值替代 getattr 兜底 (design): mgoin 接受建议，最终实现为类属性默认 False，scheduler 直接读取 coordinator 属性。
- 避免 DCP 场景误告警与冗余计算 (correctness): mgoin 认可（"Nice catch!"），最终实现将 unsupported_partial_hit_managers 计算与告警放入 if 分支内。
- SWA partial block hit 原型是否恢复 (question): 未在本次 PR 解决，作为后续演进方向（@ZJY0516 被提及）。

风险与影响

- 风险：
  1. 属性依赖风险：新逻辑要求每个 KV cache manager 正确实现 supports_fine_grained_hash_lookup；若某 manager 缺失该属性会产生 AttributeError，语义错误则会静默降级（性能损失）或错误启用（再次触发断言），建议核对所有 SingleTypeKVCacheManager 子类。

2. 命中率回退：降级后短 / 中 prompt 的 prefix 命中必须落在 scheduler_block_size 边界，PR body 已明示该 trade-off；长 prompt（如 66k token）场景实测基本无损失。
3. 行为影响面：所有含 Mamba align 组且带其他组的混合模型初始化都会走新判定；纯 full-attention 模型路径完全不变。
4. 验证局限：Kimi K3 GSM8K 实测仅 300 requests，CI 覆盖为 23 个 partial + 89 个 prefix 测试，DCP 场景未直接覆盖。 - 影响：用户 / 模型层：修复 Kimi K3 + DSpark SWA drafter 开启 prefix caching 时的断言崩溃，使这类混合模型可以安全使用前缀缓存，代价是短中 prompt 的复用可能下降。系统层：v1 引擎 KV cache 初始化与调度路径新增一次 manager 能力枚举，开销可忽略；scheduler 不再产生 partial-tail 记录，缓存布局更干净。团队层：确立了 " 按能力协商降级 " 的修复模式，为 ivanium 提到的 SWA partial block hit 原型恢复预留了清晰开关位与日志出口。 - 风险标记：核心路径变更，行为降级，依赖 manager 能力属性，验证范围有限

关联脉络

- PR #47808 [Spec Decode] DSpark confidence-scheduled verification: 本 PR 修复的正是 DSpark SWA drafter（Kimi K3 投机解码链路）与 prefix cache 的兼容冲突，属于同一功能线的后续修复。
- PR #51860 [ROCm][K3] Dequantize the fp8 decode query for MLA backends without quant-query support - TRITON_MLA: 同为 Kimi K3 投机解码链路修复，共享模型与 v1 引擎上下文。
- PR #51831 [Model] Support R3 capture with DeepGEMM MegaMoE: Kimi K3 模型支持主线之一，涉及相同模型目录与运行路径。
- PR #51311 [K3 Perf] Flash kda out kernel for prefill, 1.1~1.4x kernel performance improvement: Kimi K3 性能优化系列，同模型关注点，显示 Kimi K3 在 vLLM 中的持续投入。