PR #51614 完整报告

vllm-project/vllm

[Bugfix][KV Offload] Emit self-describing CPU events at KV-group block granularity

合并时间：2026-08-13 11:28

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51614

执行摘要

- 一句话：CPU KV 事件改为按 KV-group 块粒度发布
- 推荐动作：建议精读。该 PR 展示了一个典型的“事件发布粒度与存储索引粒度不一致”的修复，resolve_block_hashes() 在 GPU 与 CPU 事件路径的复用是值得关注的设计决策；对 KV 事件消费者实现、混合 KV cache 布局支持都有参考价值。改动集中、测试覆盖到位，合并风险低。

功能与动机

PR body 明确指出：混合 KV cache 布局中请求 hash 计算粒度（tokens_per_hash = 4）比 full-attention 组的块大小（tokens_per_block = 256）更细；原实现把每个 4-token hash 当作独立 block 发布，一个 256-token CPU chunk 会发出 64 个 4-token 的 BlockStored，与实际按 256-token 块索引的存储不一致，导致按 vLLM 块大小配置的消费者拒绝或误解这些 CPU 事件。修复目标是让 CPU 事件与 GPU 事件在块粒度上语义一致。

实现拆解

1. 变更入口：vllm/distributed/kv_transfer/kv_connector/v1/offloading/events.py 的 _build_event_metadata()，并在文件顶部 vllm.v1.core.kv_cache_utils 导入中新增 resolve_block_hashes。
2. 核心逻辑：将原先直接收集 hashes_per_chunk 个原始 hash 的逻辑，替换为调用 resolve_block_hashes(raw_chunk_hashes, tokens_per_hash, group_config.tokens_per_block)，使事件只携带每个 KV-group 块的 tail hash；block_size 由 tokens_per_hash 改为 group_config.tokens_per_block；并新增断言校验解析后的块数等于 tokens_per_chunk // tokens_per_block。
3. 保持不变：token_ids 仍保留整个 chunk；parent_block_hash 仍取 chunk 前一个原始 hash；sliding-window / SSM 组仍走 _placeholder_stored 占位路径；CPU offload key 与 chunk 查找粒度不受影响。
4. 测试配套：tests/v1/kv_connector/unit/offloading_connector/test_events.py 的 _group_config() 增加 tokens_per_hash 参数并按除法推导 hashes_per_chunk；新增参数化测试 test_event_hashes_use_group_block_size，覆盖一个 256-token 块、两个 256-token 块（512-token chunk）以及 4-token hash 粒度三种组合。

关键文件：

- vllm/distributed/kv_transfer/kv_connector/v1/offloading/events.py（模块 KV 事件；类别 source；类型 core-logic；符号 _build_event_metadata）：核心修复文件：在 _build_event_metadata() 中引入 resolve_block_hashes() 将 chunk 内 raw hash 从 tokens_per_hash 粒度解析到 KV-group 的 tokens_per_block 粒度，并修正 BlockStored.block_size 与块数量断言，是本次语义修复的关键实现单元。
- tests/v1/kv_connector/unit/offloading_connector/test_events.py（模块 事件测试；类别 test；类型 test-coverage；符号 _group_config, test_event_hashes_use_group_block_size）：测试配套文件：扩展 _group_config() 支持 tokens_per_hash 参数，新增参数化回归测试 test_event_hashes_use_group_block_size，验证 256-token 块与 4-token hash 粒度下事件 hash 与 block_size 的正确性。

关键符号：_build_event_metadata, _group_config, test_event_hashes_use_group_block_size

关键源码片段

vllm/distributed/kv_transfer/kv_connector/v1/offloading/events.py

核心修复文件：在 _build_event_metadata() 中引入 resolve_block_hashes() 将 chunk 内 raw hash 从 tokens_per_hash 粒度解析到 KV-group 的 tokens_per_block 粒度，并修正 BlockStored.block_size 与块数量断言，是本次语义修复的关键实现单元。

```
def _build_event_metadata(
    self,
    req: Request,
    group_config: 'GroupOffloadConfig',
    chunk_idx: int,
) -> _OffloadEventMetadata:
    """为一个 offloaded chunk 构建自描述事件负载。

    核心目标：把 chunk 内以 tokens_per_hash 粒度计算的原始 hash 列表，
    解析为 KV-group 的 tokens_per_block 粒度（与 GPU 侧事件保持一致）。
    """
    # 每个 chunk 内含多少个原始 hash（tokens_per_hash 粒度）
    hashes_per_chunk = group_config.hashes_per_chunk
    assert hashes_per_chunk > 0
    assert chunk_idx >= 0
    # 原始 hash 覆盖的 token 数，例如 DeepSeek V4 中为 4
    tokens_per_hash = group_config.tokens_per_chunk // hashes_per_chunk

    first_hash_idx = chunk_idx * hashes_per_chunk
    last_hash_idx = first_hash_idx + hashes_per_chunk
    assert first_hash_idx >= 0
    assert last_hash_idx <= len(req.block_hashes)

    # 取出 chunk 内的全部 raw hash，再用 resolve_block_hashes()
    # 按 KV-group 的 tokens_per_block 收敛为块粒度的 tail hash 列表。
    raw_chunk_hashes = req.block_hashes[first_hash_idx:last_hash_idx]
    chunk_hashes = resolve_block_hashes(
```

```
        raw_chunk_hashes,
        tokens_per_hash,
        group_config.tokens_per_block,
    )
    for block_hash in chunk_hashes:
        assert block_hash is not None
    # 校验解析后的块数正好等于 chunk token 数除以组块大小
    assert len(chunk_hashes) == (
        group_config.tokens_per_chunk // group_config.tokens_per_block
    )

    # 滑动窗口 / SSM 组不进此路径（调用方已过滤）
    if group_config.sliding_window_size_in_chunks is not None:
        raise AssertionError('self-describing events only support full attention')

    parent_block_hash: BlockHash | None
    if first_hash_idx == 0:
        parent_block_hash = None
    else:
        # 前一 hash 作为 parent（保持与 GPU 事件一致的链式语义）
        parent_block_hash = req.block_hashes[first_hash_idx - 1]
        assert parent_block_hash is not None

    tok_start = chunk_idx * group_config.tokens_per_chunk
    tok_end = tok_start + group_config.tokens_per_chunk
    assert tok_end <= len(req.all_token_ids)
    token_ids = tuple(req.all_token_ids[tok_start:tok_end])

    lora_id: int | None = None
    lora_name: str | None = None
    if req.lora_request is not None:
        lora_id = req.lora_request.adapter_id
        lora_name = req.lora_request.name

    return _OffloadEventMetadata(
        block_hashes=tuple(chunk_hashes),
        parent_block_hash=parent_block_hash,
        token_ids=token_ids,
        # 事件块大小改为组块大小（如 256），而非原始 hash 粒度（如 4）
        block_size=group_config.tokens_per_block,
        lora_id=lora_id,
        lora_name=lora_name,
        extra_keys=None,
        group_idx=group_config.group_idx,
        kv_cache_spec=group_config.kv_event_group_spec,
    )
```

tests/v1/kv_connector/unit/offloading_connector/test_events.py

测试配套文件：扩展 _group_config() 支持 tokens_per_hash 参数，新增参数化回归测试 test_event_hashes_use_group_block_size，验证 256-token 块与 4-token hash 粒度下事件 hash 与 block_size 的正确性。

```
def _group_config(
    *,
    group_idx: int = 0,
    block_size: int = 4,
    blocks_per_chunk: int = 1,
    tokens_per_hash: int | None = None,
    sliding_window_size_in_chunks: int | None = None,
) -> GroupOffloadConfig:
    # tokens_per_hash 缺省时与 block_size 一致，保持旧行为；
    # 传入更小粒度时可构造 hashes_per_chunk > blocks_per_chunk 的混合布局。
    if tokens_per_hash is None:
        tokens_per_hash = block_size
    tokens_per_chunk = block_size * blocks_per_chunk
    assert tokens_per_chunk % tokens_per_hash == 0
    return GroupOffloadConfig(
        group_idx=group_idx,
        tokens_per_block=block_size,
        tokens_per_chunk=tokens_per_chunk,
        hashes_per_chunk=tokens_per_chunk // tokens_per_hash,
        sliding_window_size_in_chunks=sliding_window_size_in_chunks,
        kv_event_group_spec=_FULL_ATTENTION_EVENT_SPEC,
    )


@pytest.mark.parametrize(
    ('blocks_per_chunk', 'expected_hash_indices'),
    [(1, [63]), (2, [63, 127])],
)
def test_event_hashes_use_group_block_size(
    blocks_per_chunk: int, expected_hash_indices: list[int]
):
    # 模拟 DeepSeek V4 的混合 KV 布局：
    # 原始 hash 每 4 token 计算一次，而 full-attention 组块大小为 256。
    tokens_per_hash = 4
    block_size = 256
    hashes_per_block = block_size // tokens_per_hash
    tracker = _tracker()
    group_config = _group_config(
        block_size=block_size,
        blocks_per_chunk=blocks_per_chunk,
        tokens_per_hash=tokens_per_hash,
    )
    req = _request(
        block_hashes=[_hash(i) for i in range(hashes_per_block * blocks_per_chunk)],
        token_count=block_size * blocks_per_chunk,
```

```
)
[key] = _record_chunks(tracker, req, group_config, num_chunks=1)

[event] = tracker.take_events([_stored_event([key])])

# 事件只发布每个 KV-group block 的 tail hash，且块大小等于组块大小。
assert isinstance(event, BlockStored)
assert event.block_hashes == [_wire_hash(_hash(i)) for i in expected_hash_indices]
assert event.block_size == block_size
assert len(event.token_ids) == block_size * blocks_per_chunk
```

评论区精华

orozery 作为 review 主导者多次触发 /ci run 与 /ci retry。作者确认 CI 失败与其变更无关，orozery 回复称 main 分支偶发故障，并建议更新分支或走 #pr-merge-requests 请求 force-merge。作者更新分支后触发 Buildkite CI #83512，最终 orozery 给出 APPROVED，全程无代码级 review comments。

- CI 失败排查与分支更新 (other): 作者更新分支后触发 Buildkite CI #83512，orozery 最终 APPROVED，未产生代码级 review comments。

风险与影响

- 风险：
  - 回归面有限：改动只影响开启 self_describing_kv_events 且启用 KV cache events 的 offloading 事件发布路径，默认配置下 inert。
  - 语义风险：parent_block_hash 仍直接取自 req.block_hashes[first_hash_idx - 1]，当 tokens_per_hash != tokens_per_block 时其粒度与解析后的 block_hashes 是否完全一致值得后续确认（本 PR 未改动该逻辑，推测与 GPU 侧事件保持一致）。
  - 配置假设：新断言要求 tokens_per_chunk 能被 tokens_per_block 整除，否则直接断言失败，属于显式暴露配置错误，风险可控。
  - 混合布局下不同组的 tokens_per_block 可能不同，事件消费者需按 kv_event_group_spec 区分组块大小。
  - 影响：对使用混合 KV cache 布局（如 DeepSeek V4）并开启 CPU offload 自描述事件的用户，CPU 侧事件与 GPU 侧块语义一致化，避免消费者按 256-token 块配置解析时拒绝或误解事件；对单分组场景（tokens_per_hash == tokens_per_block）行为不变；对团队而言，事件粒度语义收敛到 KV-group 块粒度，为后续新增混合布局提供更安全的事件边界。影响范围属于事件发布链路的局部修复。
  - 风险标记：parent_block_hash 粒度一致性待确认，仅影响 self-describing 事件路径，依赖 tokens_per_chunk 与 tokens_per_block 整除关系

关联脉络

- PR #51218 [Bugfix] Report FULL_ATTENTION for uniform-base UniformTypeKVCacheSpecs groups instead of UNKNOWN: 同属 self-describing KV

events / KV cache spec 语义修复，涉及混合 KV cache 组的 kind 上报，与本 PR 的事件粒度修正同一条功能线。

- PR #51843 [Bugfix] Disable fine-grained prefix-cache hits for incompatible hybrid KV layouts: 同为混合 KV 布局（hybrid KV layouts）兼容性修复，涉及 kv_cache_coordinator 与 scheduler，与本 PR 的 KV-group block 粒度问题相互补充。