

PR #53721 完整报告

vllm-project/vllm

[ROCM][Connector] SWA+HMA-support in MoRI-IO connector (Gemma4)

合并时间: 2026-09-14 20:52

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53721>

执行摘要

- 一句话: MoRI-IO 连接器支持 SWA+HMA, PD 吞吐数倍提升
- 推荐动作: 值得精读, 尤其适合关注 KV 连接器、PD 分离与 HMA 的工程师。重点看三处设计决策: 一是通过 SupportsHMA + request_finished_all_groups 把『每组一个块列表』的契约从类型层 (BlockIds) 到协议层打通, 并与 NIXL 保持一致的命名; 二是『能裁剪就裁剪、不能裁剪就 fail closed』的策略——SW 组只传窗口尾部、其他 hybrid 直接抛异常, 避免静默精度损失; 三是顺手修掉的 local/remote block id 混用 bug, 这个话题值得作为案例研究: 一个看似只关乎容量的特性改造, 如何暴露并修复既有的数据正确性问题。阅读时建议对照 test_moriiio_connector.py 中 test_get_exchange_clipped_blocks_clips_only_sw_group 与 test_metadata_hma_block_ids_preserved_per_group, 并留意 PR 未覆盖的滚动升级兼容性与多 full-attention 组场景。

功能与动机

PR body 明确写出目的: 『Run PD with sliding window with hybrid memory allocator enabled (Gemma4)』, 即在此之前 MoRI-IO 连接器在 Gemma4 这类含滑动窗口层的混合模型上无法启用 HMA, PD 分离部署只能退回 --disable-hybrid-kv-cache-manager, KV 容量受限。作者同时给出简化约定: 『READ mode only. Other hybrids (mamba etc.) need to disable HMA, as before』, 并主动声明破坏性变更: WRITE 模式的混合模型需要显式关闭 HMA, 否则抛 NotImplementedError。PR body 还附上了 GSM8K 对比、512 并发 bench serve、KV 利用率 (HMA 开约 300 并发打满 vs 关闭约 200 并发打满) 以及 DeepSeek-V3 / MiniMax-M2.5 的回归验证, 说明动机既有功能补齐也有可量化的吞吐收益。

实现拆解

1. 声明 HMA 能力并打通调度器入参: vllm/distributed/kv_transfer/kv_connector/v1/moriiio/moriiio_connector.py 中 MoRIIOConnector 由 KVConnectorBase_V1 单继承改为 (KVConnectorBase_V1, SupportsHMA); __init__ 把新入参 kv_cache_config: KVCacheConfig 透传给 MoRIIOConnectorScheduler(vllm_config, engine_id, kv_cache_config)。外层连接器新增 request_finished_all_groups(request, block_ids: BlockIds), 原 request_finished 改为把单组 block_ids 包成 (block_ids,) 后委托同一 scheduler 方法, 命名与协议对齐 NIXL 连接器 (commit 3a3ca20)。
2. 引入 BlockIds 类型统一多组语义: moriio_common.py 中 ReqMeta.local_block_ids / remote_block_ids 与 MoRIIOConnectorMetadata.add_new_req 的入参从 list[int] 改为

BlockIds (一个 KV cache 组一个列表), 连接器内部的 `_reqs_need_recv / _reqs_need_save / _reqs_need_pending_save` 也同步改成 `tuple[Request, BlockIds]`, 使多组结构在类型层面显式化。

3. Scheduler 初始化做 HMA 判定与块预算: `MoRIIOConnectorScheduler.__init__` 计算 `_is_hma_required` (未关闭 hybrid KV cache manager 且存在非 `FullAttentionSpec` 的组); 命中时先对不支持的 hybrid 类型 (只允许 `FullAttentionSpec + SlidingWindowSpec`) fail closed 抛 `NotImplementedError`, 再对 WRITE 模式抛 `NotImplementedError` (PR body 声明的 breaking change); 随后计算 `blocks_per_sw = cdiv(sliding_window, block_size) + 1` (+1 保守兜底窗口与块边界不对齐), 并定位第一个 full-attention 组, 记录 `_full_attn_group_idx / _full_attn_block_size` 作为 chunked prefill 的 token 计数基准, 避免用被裁剪的 SW 组块数推断序列长度。
4. 新增裁剪逻辑并接入两条交换路径: `get_exchange_clipped_blocks(block_ids)` 在非 HMA 时原样返回 (`[list(blocks) for blocks in block_ids]`), HMA 时对 `blocks_per_sw[i] > 0` 的组只保留窗口尾部块、full-attention 组全量保留。`update_state_after_alloc` 的 save 路径改为整体保存 `blocks.get_block_ids()`; `recv` 路径用裁剪后的真实本地块替换旧实现中『`local_block_ids = remote_block_ids[-len(local_block_ids):]`』的覆盖写法, 并把长度断言从全局比较改为逐组比较 (前缀缓存部分命中时本地块可少于远端块, 但不会更多)。
5. 修复 local/remote block id 混用回归: 作者在 review 中自曝该 bug, 修复集中在 commit f8d28306, GSM8K strict-match 由 0.7278 (1-2p 回归) 恢复到 0.7362, 接近 main 的 0.7392。
6. 测试与验证配套: `tests/v1/kv_connector/unit/test_moriio_connector.py` 新增 `_make_hybrid_kv_cache_config` (full attention + `SlidingWindowSpec` 各一组)、`_read_scheduler` 夹具, 以及 `test_hma_blocks_per_sw_two_groups`、`test_is_hma_required` (三种参数化)、`test_non_sliding_window_hybrid_is_rejected`、`test_token_count_basis_uses_full_attention_group`、`test_get_exchange_clipped_blocks_clips_only_sw_group`、`test_metadata_hma_block_ids_preserved_per_group`、`test_single_group_path_unchanged`、`test_hybrid_write_mode_rejected`、`test_worker_layer_to_group_routing` 等用例, 且把既有断言改为 `req_meta.local_block_ids[0]` 以适配新结构; `tests/v1/kv_connector/unit/test_moriio_routing.py` 的 mock producer 补上 `_is_hma_required=False`、`blocks_per_sw=[0]` 并把 `get_exchange_clipped_blocks` 绑定到 stand-in 对象, 避免 `SimpleNamespace` 绕过新逻辑导致测试失真。

关键文件:

- `vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py` (模块 KV 连接器; 类别 source; 类型 core-logic; 符号 `MoRIIOConnector`, `MoRIIOConnector.request_finished`, `MoRIIOConnector.request_finished_all_groups`, `MoRIIOConnectorScheduler.init`): 本 PR 的绝对核心: `MoRIIOConnector` 继承 `SupportsHMA`、新增 `request_finished_all_groups`; `MoRIIOConnectorScheduler` 接收 `KVCacheConfig` 并计算 `_is_hma_required / blocks_per_sw / full-attention` 基准组; 新增 `get_exchange_clipped_blocks` 并在 `update_state_after_alloc` 中修掉 local/remote block id 混用的回归。

- tests/v1/kv_connector/unit/test_moriio_connector.py (模块 连接器测试; 类别 test; 类型 test-coverage; 符号 _make_hybrid_kv_cache_config, _read_scheduler, test_hma_blocks_per_sw_two_groups, test_is_hma_required) : 新增 Gemma-like 的 hybrid KV cache 夹具与 10 个左右用例, 覆盖 HMA 判定、blocks_per_sw 计算、SW 组裁剪、非 SW hybrid 拒绝、WRITE+HMA 拒绝、多组块 ID 保留等关键分支, 并把既有断言改为 local_block_ids[0] 以适配新的多组结构。
- vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_common.py (模块 元数据类型; 类别 source; 类型 data-contract; 符号 ReqMeta, MoRIIOConnectorMetadata.add_new_req, BlockIds) : 把 ReqMeta 的 local_block_ids / remote_block_ids 与 add_new_req 入参从 list[int] 改为 BlockIds (list[list[int]]), 是连接器内部多组块语义的数据契约变更, 连接器与 worker 两侧的类型约定都依赖它。
- tests/v1/kv_connector/unit/test_moriio_routing.py (模块 路由测试; 类别 test; 类型 test-coverage; 符号 _producer) : DP-rank 路由测试的 mock producer 需要补齐 _is_hma_required=False、blocks_per_sw=[0] 并绑定 get_exchange_clipped_blocks, 否则 request_finished 路径会因为缺少 HMA 属性而失败, 属于源码契约变化引发的测试夹具适配。

关键符号: MoRIIOConnector.init, MoRIIOConnector.request_finished, MoRIIOConnector.request_finished_all_groups, MoRIIOConnectorScheduler.init, MoRIIOConnectorScheduler.get_exchange_clipped_blocks, MoRIIOConnectorScheduler.update_state_after_alloc, MoRIIOConnectorMetadata.add_new_req

关键源码片段

tests/v1/kv_connector/unit/test_moriio_connector.py

新增 Gemma-like 的 hybrid KV cache 夹具与 10 个左右用例, 覆盖 HMA 判定、blocks_per_sw 计算、SW 组裁剪、非 SW hybrid 拒绝、WRITE+HMA 拒绝、多组块 ID 保留等关键分支, 并把既有断言改为 local_block_ids[0] 以适配新的多组结构。

测试用最小化的 Gemma-like 配置把『一个 full-attention 组 + 一个 sliding-window 组』固定下来, 并直接断言裁剪结果:

```
def _make_hybrid_kv_cache_config() -> KVCacheConfig:
    """一个 full-attention 组 + 一个 sliding-window 组 (模拟 Gemma) 。”
    full_spec = FullAttentionSpec(
        block_size=16, num_kv_heads=4, head_size=64, dtype=torch.float16
    )
    sw_spec = SlidingWindowSpec(
        block_size=16,
        num_kv_heads=4,
        head_size=64,
        dtype=torch.float16,
        sliding_window=32, # window=32、block=16 → 期望交换 3 个块
    )
    num_blocks = 2
```

```

page = full_spec.page_size_bytes
return KVCacheConfig(
    num_blocks=num_blocks,
    kv_cache_tensors=[
        KVCacheTensor(
            size=2 * num_blocks * page,
            layers=["full0", "sw0"],
            layer_stride=num_blocks * page,
            block_stride=page,
        )
    ],
    kv_cache_groups=[
        KVCacheGroupSpec(layer_names=["full0"], kv_cache_spec=full_spec),
        KVCacheGroupSpec(layer_names=["sw0"], kv_cache_spec=sw_spec),
    ],
)

```

```

def _read_scheduler(
    kv_cache_config: KVCacheConfig, disable_hma: bool = False
) -> MoRIIOConnectorScheduler:
    vllm_config = create_vllm_config(role="kv_producer", read_mode=True)
    # disable_hma 对应命令行 --disable-hybrid-kv-cache-manager 的配置项,
    # 用于验证关闭 HMA 时 _is_hma_required 与裁剪逻辑都退化为 no-op。
    vllm_config.scheduler_config.disable_hybrid_kv_cache_manager = disable_hma
    with set_current_vllm_config(vllm_config):
        connector = MoRIIOConnector(
            vllm_config, KVConnectorRole.SCHEDULER, kv_cache_config
        )
        assert connector.connector_scheduler is not None
    return connector.connector_scheduler

```

```

def test_get_exchange_clipped_blocks_clips_only_sw_group():
    """裁剪只作用于 sliding-window 组, full attention 组必须原样保留。"""
    scheduler = _read_scheduler(_make_hybrid_kv_cache_config())
    full = [10, 11, 12, 13, 14]
    sw = [20, 21, 22, 23, 24]
    clipped = scheduler.get_exchange_clipped_blocks([full, sw])
    assert clipped[0] == full # full attention 组: 5 个块全部保留
    assert clipped[1] == [22, 23, 24] # SW 组: cdiv(32, 16) + 1 = 3, 只留尾部

```

评论区精华

1) 作者自曝并自证的正确性回归 (moriio_connector.py:663) : simondanielsson 在 `update_state_after_alloc` 处写道 > "This was a bug: `local_block_ids = remote_block_ids[...]` where we need these local ids to be truly the local ones", 随后用数据补强: Before this change (1-2p 回归) flexible-extract 0.7369 / strict-match 0.7278

; After removing this local->remote block id override (commit f8d28306) 0.7422 / 0.7362; Compare to main (TP1 无 PD) 0.7468 / 0.7392, 确认回归消除。这是本 PR 最有价值的讨论, 说明 HMA 改造顺带暴露了旧代码里用远端块 ID 冒充本地块 ID 的语义错位。

2) 代码组织建议被采纳: dllehr-amd 针对新增的 HMA 检查块留下 > "can we move this under line 409's check?", 作者在 commit [ea18bb01ff73a7733a444ba1f87178812939b720](#) (chore: move check for HMA+WRITE mode into the same if statement) 中把 WRITE 模式检查合入 `_is_hma_required` 分支, 减少一层独立判断。3) fork PR 缺少自动 review: claude[bot] 说明 > "This pull request is from a fork — automated review is disabled. A repository maintainer can comment [@claude review](#) to run a one-time review.", 实际上主要依赖作者自审 + tjtaanaa 的 APPROVED, review_comments 仅 3 条。4) CI 排队与重试: AndreasKaratzas 取消 AMD 构建 > "I cancelled the build here. Let's retrigger it later when the queues are decongested first", 作者在 5 个 commit 上反复 `/ci run`、`/ci retry`、`/amd-ci retry`, 说明该 PR 的合入主要卡在 AMD/ 上游 CI 资源而非代码争议。

- `update_state_after_alloc` 用 `remote_block_ids` 覆盖 `local_block_ids` 的正确性回归 (correctness): 确认为真实 bug, 已在 commit f8d28306 中改为使用 `blocks.get_block_ids()` 经 `get_exchange_clipped_blocks` 裁剪后的真实本地块, 并把断言改为逐组比较本地块数不超过远端块数。
- HMA + WRITE 模式检查的代码位置 (design): 作者采纳, 见 commit [ea18bb01ff73a7733a444ba1f87178812939b720](#) (chore: move check for HMA+WRITE mode into the same if statement)。
- fork PR 缺少自动化 review 覆盖 (testing): 未强制要求额外人工深审, 代码正确性主要依赖作者自测 (GSM8K、512 并发 bench、DeepSeek-V3 / MiniMax-M2.5 回归)。
- CI 排队拥塞导致的构建取消与重跑 (other): 合入主要阻塞在 CI 资源而非代码分歧, 最终以 [b6275a5bce8e](#) 为 head 通过并 merge。

风险与影响

- 风险: 1) 破坏性行为变更 (高): `MoRIIOConnectorScheduler.__init__` 对 Hybrid 模型 + WRITE 模式直接抛 `NotImplementedError`, 错误信息要求用户改用 READ 模式或传 `--disable-hybrid-kv-cache-manager`。已按此方式部署的存量脚本在升级后会启动失败, 属于显式 breaking change, 需要在 release note / 文档中标注。2) fail closed 范围扩大 (中): 只放行 `FullAttentionSpec` + `SlidingWindowSpec`, `ChunkedLocalAttentionSpec`, `mamba` 等一律拒绝启动。虽然没有静默传错块的风险 (`test_non_sliding_window_hybrid_is_rejected` 已覆盖), 但把此前『只是不启用 HMA 也能跑』的场景变成硬失败, 影响面比标题 (Gemma4) 所暗示的更大。3) 块预算保守带来的额外传输 (低 - 中): `blocks_per_sw = cdiv(window, block_size) + 1` 的 +1 是为窗口与块边界不对齐留的余量, 窗口远大于 `block_size` 时会多传一个块; 这是作者有为之的保守设计 (代码注释写明), 但对 RDMA 传输量有轻微影响, 未被 bench 单独量化。4) 块 ID 语义修复的覆盖边界 (中): `update_state_after_alloc` 中改为逐组 `assert len(local) <= len(remote)`, 若两端组数或块数不满足预期会直接断言失败而非降级处理; 同时 `remote_block_ids` 现在按组传递, `prefill` 与 `decode` 滚动升级、版本不一致时协议可能不兼容。5) 验证证据偏手工 (中): 全部性能与精度数据来自作者在 MI300 上的手工运行, 仓库内没有新增 e2e/CI 用例 (MoRI-IO 依赖专有 `mori.io` 库, CI 无法覆盖), 单测大量使用 `mock/stand-in`, 无法覆盖

真实 HMA 数据面。6) full-attention 基准组的选择（低）：`_full_attn_group_idx` 取第一个 `blocks_per_sw == 0` 的组，若模型存在多个 `block_size` 不同的 full-attention 组，token 计数基准未必唯一，当前仅在 Gemma-like 两组件下被测。

- 影响：对用户：Gemma4 等 SW 混合模型在 ROCm + MoRI-IO 的 PD 分离部署下可以打开 HMA，PR body 实测 512 并发时 QPS 由 0.27 提升到 1.10、Mean TPOT 由 860.62ms 降到 240.80ms，KV 打满时的并发请求数由约 200 升至约 300，精度与 main 持平（GSM8K flexible-extract 0.7422 vs 0.7468）。但使用 Hybrid 模型 + WRITE 模式的用户升级后会立即失败，必须改配置。对系统：影响集中在 `kv_connector/v1/moriiio` 路径与调度器的块交换语义，不改动 attention 内核或其他连接器；`BlockIds` 类型的引入让多组块列表成为连接器内部的数据契约，后续 worker 侧 `_write_blocks_for_req`、层到组映射等逻辑都会围绕它展开，是 MoRI-IO 向 HMA 演进的结构化一步。对团队：补齐了 MoRI-IO 相对 NIXL 在 HMA 能力上的缺口（函数命名也对齐 NIXL），维护者后续需要同步 mamba 等其他 hybrid 的处置策略；验证仍依赖 AMD 硬件与手工脚本，短期内该路径的回归风险由作者与 AMD 侧承担。
- 风险标记：核心路径变更，WRITE 模式破坏性变更，fail-closed 拒绝其他 hybrid，端到端验证依赖手工 AMD 机器，缺少滚动升级兼容性说明

关联脉络

- PR #45043 MoRI-IO router-authoritative DP-rank 路由重设计（njhill review 反馈，标题未在上下文中给出）：`tests/v1/kv_connector/unit/test_moriiio_routing.py` 的文件头注释明确写这是为落实 PR #45043 的 review 反馈而做的重设计；本 PR 修改的正是该测试文件（补 `_is_hma_required / blocks_per_sw` 并绑定 `get_exchange_clipped_blocks`），属同一条 MoRI-IO 路由与块交换演进线。
- PR #50984 [Bugfix][Mooncake] Report failed remote KV loads to the scheduler: 同属 vllm v1 分层 KV 连接器的可靠性修复：Mooncake 让解码侧上报远端加载失败以解除请求滞留，本 PR 则修正了 READ 路径上本地 / 远端块 ID 的语义错位，两者都是远端 KV 获取路径上的正确性问题，可对照阅读。
- PR #56317 [Bugfix][NixlPush] Guard `_remote_agents read in _do_send_reg_notif` (X1): NIXL 连接器是 SupportsHMA 与多组 block ids 的既有参考实现（本 PR 的 commit 3a3ca20 明确『rename new functions to align with NIXL connector』），两者的调度器 / 工作线程分层的块交换接口可互为参照。