PR #33308 完整报告

sgl-project/sglang

[Fix] Drop deprecated multimodal processor residency state

合并时间：2026-08-03 11:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33308

执行摘要

- 一句话：移除废弃 mm 驻留状态，以 use_cuda_ipc 统一设备驻留
- 推荐动作：值得精读。重点看三处设计： 1) 废弃参数清理时如何把状态所有权收敛到 use_cuda_ipc 单一来源； 2) _precompute_hashes_before_cpu_transfer 的"GPU 侧先哈希、再统一搬 CPU"模式，对依赖哈希做前缀缓存的模型很有价值； 3) EPD 失败不降级而是返回 503 的防护思路。同时可学习其对客户端 / 服务端错误分类的测试设计。

功能与动机

PR body 明确指出：--keep-mm-feature-on-device is deprecated and _handle_multimodal_feature_transport already forces it to False, so the processor-side reads are dead. Remove them and let use_cuda_ipc own device residency. 即处理器端对该参数的读取是死代码，删除后应由 use_cuda_ipc 统一决定设备驻留；同时 ernie45_vl / midashenglm 仍读取该属性，必须同步修复。另外，IPC pool-handle 缓存此前是 import 时的一次性环境快照，无法在运行时（尤其测试中）灵活控制，因此改为实例级标志。Kimi K2.5 等模型需要在 GPU 上先基于特征计算 hash / pad_value 再搬到 CPU，这催生了 precompute_hash_before_cpu_transfer 扩展点。

实现拆解

1. 清理基类废弃状态（base_processor.py）：删除模块级 _IPC_POOL_HANDLE_CACHE = envs.SGLANG_USE_IPC_POOL_HANDLE_CACHE.get() 快照和 self.keep_mm_feature_on_device = server_args.keep_mm_feature_on_device；新增实例级 self.use_ipc_pool_handle_cache = self.use_cuda_ipc and envs.SGLANG_USE_IPC_POOL_HANDLE_CACHE.get()，_wrap_tensor_for_cuda_ipc 改用它决定是否附加 pool_ipc_handle。这一步让设备驻留与 IPC 缓存行为全部收敛到实例配置，消除废弃参数与 import 时快照的隐患。
2. 新增延迟 CPU 转移与哈希预计算（base_processor.py）：process_mm_data 中的转移条件由 if not self.keep_mm_feature_on_device 改为 if not self.use_cuda_ipc and not self.precompute_hash_before_cpu_transfer，拆掉原有的 if self.use_cuda_ipc: pass 空分支；新增类属性 prefer_tokenized_input = False、precompute_hash_before_cpu_transfer = False；新增静态方法 _move_feature_to_cpu（递归处理 Tensor / list / tuple）与实例方法 _precompute_hashes_before_cpu_transfer（先对每个 MultimodalDataItem 调 set_pad_value() 在 GPU 侧算 hash，再统一搬到

CPU）；process_and_combine_mm_data 在进入 cuda_ipc 后处理前调用该方法。

3. 模型处理器修复与扩展点启用（ernie45_vl.py / midashenglm.py / kimi_k25.py / schedule_batch.py）：ernie45_vl 与 midashenglm 的 CPU 迁移条件从 not self.keep_mm_feature_on_device 改为 not self.use_cuda_ipc；kimi_k25 的 KimiK2_5VLImageProcessor 开启 prefer_tokenized_input = True 与 precompute_hash_before_cpu_transfer = True；MultimodalDataItem 新增 set_hash(hash_value)，同时写入 hash 与 pad_value。
4. tokenizer_manager 增强（tokenizer_manager.py）：新增 _reject_missing_dispatched_encoder_embedding，在 language_only + zmq_to_tokenizer + need_wait_for_mm_inputs 且未收到 mm_inputs 时直接抛 503，避免把 encoder 失败静默降级为本地视觉处理；_tokenize_one_request 中按 self.mm_processor.prefer_tokenized_input 选择把 input_ids 还是文本传给处理器；caller mm_hashes 写入从 item.hash = int(...) 改为 item.set_hash(...)，保证 pad_value 同步。
5. 错误分类与测试配套：fast_load_mm_data / legacy_load_mm_data 中 ValueError 不再被包装成 RuntimeError，而是作为客户端错误原样上抛（日志降为 info），其他异常仍包装为 RuntimeError（服务端错误）；测试沿用此语义。配套测试更新 test_mm_process_config.py（改用 CustomTestCase、删除废弃属性 mock、新增 test_cuda_ipc_pool_handle_cache_can_be_disabled 与 TestPrecomputeHashBeforeCpuTransfer）和 test_base_processor_image_decode.py（新增 4 个错误分类回归测试）。

关键文件：

- python/sglang/srt/multimodal/processors/base_processor.py（模块 处理器基类；类别 source；类型 core-logic；符号 _move_feature_to_cpu, _precompute_hashes_before_cpu_transfer, use_ipc_pool_handle_cache, prefer_tokenized_input）：核心改动：删除 keep_mm_feature_on_device 读取，IPC 缓存实例化，新增 prefer_tokenized_input / precompute_hash_before_cpu_transfer 及 _precompute_hashes_before_cpu_transfer / _move_feature_to_cpu。
- test/registered/unit/managers/test_mm_process_config.py（模块 配置测试；类别 test；类型 test-coverage；符号 TestMmProcessConfigValidation, TestBaseProcessorConfigExtraction, TestMultimodalFeatureTransportRuntime, test_cuda_ipc_pool_handle_cache_can_be_disabled）：测试同步：移除废弃属性 mock，新增 IPC handle 缓存禁用测试与哈希预计算测试，改用 CustomTestCase 并去掉 AMD 注册。
- python/sglang/srt/managers/tokenizer_manager.py（模块 请求预处理；类别 source；类型 core-logic；符号 _reject_missing_dispatched_encoder_embedding, _tokenize_one_request）：新增 EPD 失败 503 防护 _reject_missing_dispatched_encoder_embedding，支持 prefer_tokenized_input 直传 input_ids，并将 caller mm_hashes 改走 MultimodalDataItem.set_hash。
- test/registered/unit/multimodal/test_base_processor_image_decode.py（模块 图像解码；类别 test；类型 test-coverage；符号 test_fast_loader_preserves_invalid_input_as_va

lue_error, test_unreachable_image_url_is_a_client_error,test_invalid_image_bytes_are_a_client_error,test_unexpected_loader_bug_remains_a_server_error）：新增 4 个错误分类测试，验证ValueError 作为客户端错误、其他异常作为服务端错误的语义。

- python/sglang/srt/multimodal/processors/ernie45_vl.py（模块 模型处理器；类别 source；类型 core-logic；符号 process_mm_data）：修复仍读取废弃 keep_mm_feature_on_device 的问题，改用 use_cuda_ipc 决定是否将特征搬到 CPU。
- python/sglang/srt/managers/schedule_batch.py（模块 批次调度；类别 source；类型 core-logic；符号 set_hash）：MultimodalDataItem 新增 set_hash，同时设置 hash 与 pad_value，供外部路由 hash 使用。
- python/sglang/srt/multimodal/processors/kimi_k25.py（模块 模型处理器；类别 source；类型 core-logic；符号 KimiK2_5VLImageProcessor）：KimiK2_5VLImageProcessor 开启 prefer_tokenized_input 与 precompute_hash_before_cpu_transfer，启用新扩展点。
- python/sglang/srt/multimodal/processors/midashenglm.py（模块 模型处理器；类别 source；类型 core-logic；符号 process_mm_data）：修复仍读取废弃 keep_mm_feature_on_device 的问题，改用 use_cuda_ipc。

关键符号：_move_feature_to_cpu, _precompute_hashes_before_cpu_transfer, _reject_missing_dispatched_encoder_embedding, set_hash, process_mm_data, process_and_combine_mm_data, fast_load_mm_data, legacy_load_mm_data

关键源码片段

python/sglang/srt/multimodal/processors/base_processor.py

核心改动：删除 keep_mm_feature_on_device 读取，IPC 缓存实例化，新增 prefer_tokenized_input / precompute_hash_before_cpu_transfer 及 _precompute_hashes_before_cpu_transfer / _move_feature_to_cpu。

```
# python/sglang/srt/multimodal/processors/base_processor.py

class BaseMultimodalProcessor(ABC):
    models = []
    gpu_image_decode = True # 默认启用 GPU 解码
    # 新增处理器扩展点：prefer_tokenized_input 表示该处理器更愿意接收
    # token id 而非原始文本；precompute_hash_before_cpu_transfer 表示
    # 需要在把特征搬到 CPU 之前，先在 GPU 上计算 hash 与 pad_value。
    prefer_tokenized_input = False
    precompute_hash_before_cpu_transfer = False

    def __init__(self, hf_config, server_args, _processor, transport_mode, *args, **kwargs):
        ...
        # 设备驻留的唯一所有权归 use_cuda_ipc：不再读取已废弃的
        # keep_mm_feature_on_device。
        self.mm_feature_transport = (
            configured_mm_feature_transport
            if configured_mm_feature_transport in ("cpu", "cuda_ipc")
            else "cpu"
```

```
        )
        self.use_cuda_ipc = self.mm_feature_transport == "cuda_ipc"
        # IPC pool handle 缓存改为实例级标志：结合当前 ServerArgs 与运行时
        # env 实时判定，而不是 import 时的一次性快照。
        self.use_ipc_pool_handle_cache = (
            self.use_cuda_ipc and envs.SGLANG_USE_IPC_POOL_HANDLE_CACHE.get()
        )
        ...

    @staticmethod
    def _move_feature_to_cpu(value):
        # 递归地把 Tensor / list / tuple 中的特征统一搬到 CPU；
        # 非 Tensor 元素（例如 dict 元数据）原样保留。
        if isinstance(value, torch.Tensor):
            return value.cpu()
        if isinstance(value, list):
            return [BaseMultimodalProcessor._move_feature_to_cpu(v) for v in value]
        if isinstance(value, tuple):
            return tuple(BaseMultimodalProcessor._move_feature_to_cpu(v) for v in value)
        return value

    def _precompute_hashes_before_cpu_transfer(self, mm_items):
        # 在特征迁移到 CPU 之前，先在 GPU 上为每个 item 计算 hash 与
        # pad_value（set_pad_value 内部会 hash_feature 并写 pad_value），
        # 再递归把 feature 与 precomputed_embeddings 搬回 CPU，避免
        # 在 CPU 侧重复计算哈希导致语义漂移。
        if not self.precompute_hash_before_cpu_transfer:
            return
        for item in mm_items:
            item.set_pad_value()
            if not self.use_cuda_ipc:
                item.feature = self._move_feature_to_cpu(item.feature)
                item.precomputed_embeddings = self._move_feature_to_cpu(
                    item.precomputed_embeddings
                )
```

python/sglang/srt/managers/tokenizer_manager.py

新增 EPD 失败 503 防护 _reject_missing_dispatched_encoder_embedding，支持 prefer_tokenized_input 直传 input_ids，并将 caller mm_hashes 改走 MultimodalDataItem.set_hash。

```
# python/sglang/srt/managers/tokenizer_manager.py

def _reject_missing_dispatched_encoder_embedding(server_args, request_obj, mm_inputs):
    # 防止 EPD 请求在缺少 encoder embedding 时被静默降级为本地视觉处理。
    # 在 language_only 且走 zmq_to_tokenizer 传输、请求又明确要求等待
    # encoder embedding 的情况下，如果没有收到 mm_inputs，直接返回 503，
    # 避免把「encoder 失败」悄悄变成「本地再算一次」的隐性错误。
    if (
```

```
            mm_inputs is None
            and server_args.language_only
            and server_args.encoder_transfer_backend == "zmq_to_tokenizer"
            and request_obj.need_wait_for_mm_inputs
        ):
            raise fastapi.HTTPException(
                status_code=HTTPStatus.SERVICE_UNAVAILABLE,
                detail=(
                    "The encoder did not return multimodal embeddings. "
                    "The request was not run locally in language-only mode."
                ),
            )
```

python/sglang/srt/managers/schedule_batch.py

MultimodalDataItem 新增 set_hash，同时设置 hash 与 pad_value，供外部路由 hash 使用。

```
# python/sglang/srt/managers/schedule_batch.py 中 MultimodalDataItem 的新增方法

def set_hash(self, hash_value: int) -> None:
    # 由外部（如调用方提供的 mm_hashes）直接写入 hash，并同步算出
    # pad_value，保证路由决策与 sglang 前缀缓存键一致。
    self.hash = hash_value
    self.pad_value = _compute_pad_value(hash_value)
```

评论区精华

本 PR 没有 reviewer 评论（review_comments_count = 0）。作者通过 issue 评论 /rerun-test 触发了 8 组定向测试（mm process config、base processor image decode、kimi_k25、cuda_ipc transport、cuda_ipc pool budget、EPD disaggregation、vision openai server），github-actions 回报全部通过；随后 /tag-and-rerun-ci 进入全量 CI。唯一非作者评论是 gemini-code-assist 的停用通知，无实质内容。

- CI 重跑与测试验证 (testing): 定向测试全部通过，随后 /tag-and-rerun-ci 进入全量 CI。

风险与影响

- 风险：
  1. EPD 失败路径行为变更（tokenizer_manager.py）：_reject_missing_dispatched_encoder_embedding 在 language_only + zmq_to_tokenizer + need_wait_for_mm_inputs 且未收到编码结果时直接返回 503；此前是打 warning 后静默降级为本地处理。依赖旧降级语义的调用方会收到 503，属于有意的语义收紧，但需确认对外 API 兼容性。
  2. 新增哈希预计算路径（base_processor.py / kimi_k25.py）：_precompute_hashes_before_cpu_transfer 目前仅 Kimi K2.5 开启。该路径要求 item 必须有可哈希的 feature 或 precomputed_embeddings，否则 set_pad_value() 内的 assert self.hash is not None 会失败；若 Kimi K2.5 某些输入分支缺失特征，可能引发新回归。

3. set_hash 提前计算 pad_value（schedule_batch.py）：caller mm_hashes 现在立即同步 pad_value，此前 pad_value 由后续 set_pad_value() 惰性计算；哈希一致时 pad_value 也应一致，但任何依赖惰性语义的隐式路径需留意。
4. IPC 缓存实例化（base_processor.py）：移除模块级 _IPC_POOL_HANDLE_CACHE 后，行为改为随 env 实时变化；仓库内已无外部引用，但若有进程内复用模块且期望 import 时快照的外部代码会受影响。
5. 错误分类变化（base_processor.py）：多模态加载时的 ValueError 不再包装为 RuntimeError，依赖"所有加载异常都是 500"的客户端监控会看到状态码变化（4xx vs 5xx）。 - 影响：用户侧：无直接参数变化（keep_mm_feature_on_device 早已被强制为 False），但 EPD / language_only + zmq_to_tokenizer 场景下，encoder 失败从静默降级变为明确 503，错误暴露更可观测；Kimi K2.5 的多模态请求改走 tokenized input 与 GPU 侧哈希预计算，前缀缓存键与路由决策更一致。系统侧：多模态特征生命周期更加清晰，设备驻留单一由 use_cuda_ipc 决定；IPC pool handle 缓存可实例级控制，便于测试与调优；错误分类统一为客户端 ValueError / 服务端 RuntimeError。团队侧：新增 prefer_tokenized_input 与 precompute_hash_before_cpu_transfer 两个处理器扩展点，后续模型接入可复用同一套机制。 - 风险标记：EPD 失败路径行为变更，新增哈希预计算路径，Kimi K2.5 输入模式切换，错误分类影响监控

关联脉络

- PR #30177 [Feature] Support return_hidden_states="last": 与本 PR 都改动了 schedule_batch.py 及多模态数据链路，同属多模态特征生命周期管理演进。
- PR #33025 [Kimi K3] Add reasoning, tool-call, and OpenAI serving support: 同属 Kimi 模型家族处理器（kimi_k25）与多模态处理链路，本 PR 为 K2.5 开启了新的 tokenized input 与哈希预计算模式。
- PR #33294 test: stand up the config tiers two unit tests read from: 同为多模态相关测试配置与夹具的维护，反映 mm 处理配置持续收紧的趋势。