

NPU 驱动和运行库开发参考

文档版本 V1.4

发布日期 2025-12-15

前言

概述

本文用于指导开发人员基于现有模型，使用驱动层和运行层提供的 C 语言 API 库开发深度神经网络应用，用于实现目标识别、图像分类等功能。

通过本文档您可以达成：

- 了解驱动层和运行层的功能架构，基本概念以及接口的使用流程
- 使用驱动层和运行层接口进行应用开发的基本流程和实现方法
- 能够基于本文档的使用流程，扩展进行其他应用的开发



说明

未经特殊说明，以 XM7606V13 指代 XM7606 系列所有芯片，以 XM7206V11A 指代 XM7206 系列所有芯片。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不可避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不可避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不可避免则可能导致轻微或中度伤害的具有低等级风险的危害。
 须知	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。
2025-06-05	V1.1	第 3 章 运行时 API 参考 3.2.1 小节，函数说明增加

修订日期	版本	修订说明
		xmedia_cl_profiling_type_e、 xmedia_cl_profiling_flag_e、 xmedia_cl_profiling_params。 新增 3.2.23、3.2.24、3.2.25 小节。 3.2.26 小节，新增-74，-75 错误码。 第 5 章 运行时用户使用流程 新增有关动态 batch、设置回调函数、 profiling 工具的相关描述。
		第 3 章 运行时 API 参考 3.2.1 小节，函数说明增加 xmedia_cl_set_workspace_addr、 xmedia_cl_get_workspace_addr、 xmedia_cl_graph_get_private_data。 新增 3.2.26、3.2.27、3.2.28 小节。 3.2.29 小节，新增-76、-77 错误码。
2025-08-20	V1.2	第 3 章 运行时 API 参考 3.2.1 小节，函数说明增加 xmedia_cl_memory_reuse_type。 新增 3.2.29 小节。 第 5 章 运行时用户使用流程 新增有关内存复用描述。
		第 2 章 UMD API 参考 2.1 小节，修改 xmedia_npu_open_params_s 与 xmedia_npu_start_params_s。 第 3 章 运行时 API 参考 3.2.4、3.2.12、3.2.13、3.2.16、3.2.24、 3.2.26、3.2.29 小节新增函数约束说明。
2025-09-29	V1.3	3.2.1 小节，修改 xmedia_cl_data_type。

修订日期	版本	修订说明
2025-12-15	V1.4	第 2 章 UMD API 参考 2.1 小节，更新 UMD 函数参数说明。 2.2 小节，更新 UMD API 接口。
		第 3 章 运行层 API 参考 3.2.1 小节，修改 xmedia_cl_data_type， 新增 xmedia_cl_mem_info、 xmedia_cl_model_info_type 函数参数说明。 3.2.6 小节，更新返回值说明。 3.2.26 小节，新增约束说明。 新增 3.2.30、3.2.31、3.2.32、3.2.33、 3.2.34、3.2.35、3.2.36、3.2.37 小节。 3.2.38 小节，新增-78、-79 错误码。

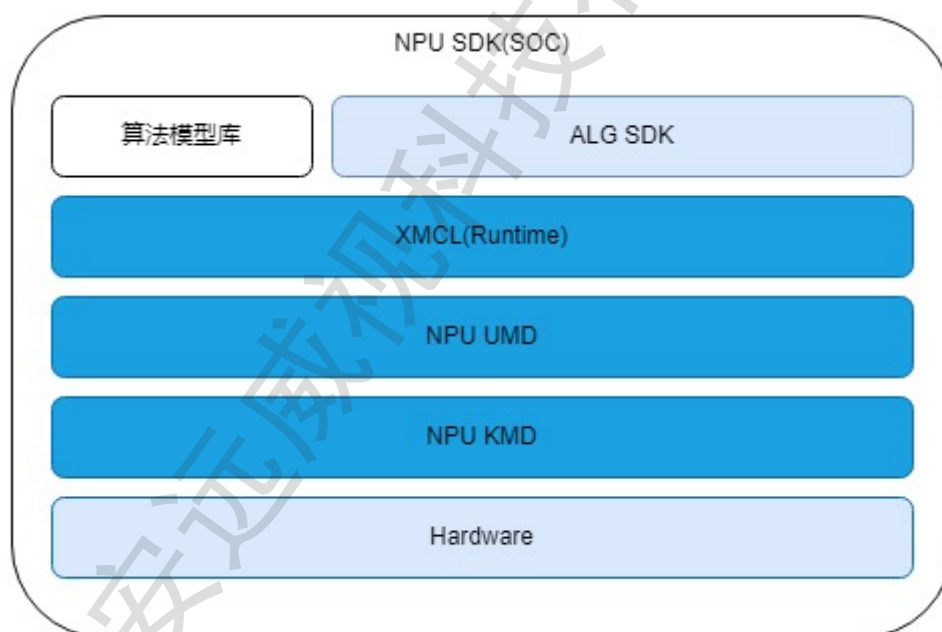
目 录

1 概述.....	1
1.1 软件栈.....	1
1.2 Feature List	1
1.3 基本介绍	2
2 UMD API 参考.....	3
2.1 函数参数说明.....	3
2.2 API 接口.....	3
2.2.1 xmedia_npu_get_usage_rate	3
2.2.2 xmedia_npu_get_proc_info	3
2.2.3 xmedia_npu_set_quick_start_flag.....	4
2.2.4 xmedia_npu_get_quick_start_flag.....	4
3 运行层 API 参考	5
3.1 名词解释.....	5
3.2 运行时 API.....	5
3.2.1 函数参数说明	5
3.2.2 xmedia_cl_init.....	10
3.2.3 xmedia_cl_uninit.....	10
3.2.4 xmedia_cl_get_device_ids	10
3.2.5 xmedia_cl_release_device_ids.....	11
3.2.6 xmedia_cl_create_context.....	11
3.2.7 xmedia_cl_release_context.....	12
3.2.8 xmedia_cl_graph_querysize_from_file.....	12
3.2.9 xmedia_cl_graph_querysize_from_buff	12

3.2.10 xmedia_cl_graph_loadmodel_from_file	13
3.2.11 xmedia_cl_graph_loadmodel_from_buff	14
3.2.12 xmedia_cl_graph_loadmodel_from_file_withmem	14
3.2.13 xmedia_cl_graph_loadmodel_from_buff_withmem	15
3.2.14 xmedia_cl_graph_get_input	16
3.2.15 xmedia_cl_graph_get_output	16
3.2.16 xmedia_cl_graph_set_inout	17
3.2.17 xmedia_cl_graph_process	17
3.2.18 xmedia_cl_graph_submit	18
3.2.19 xmedia_cl_graph_unload	18
3.2.20 xmedia_cl_wait_for_events	19
3.2.21 xmedia_cl_query_event_status	19
3.2.22 xmedia_cl_release_event	19
3.2.23 xmedia_cl_set_event_callback	20
3.2.24 xmedia_cl_graph_set_dynamic_batch_size	20
3.2.25 xmedia_cl_set_profiling	21
3.2.26 xmedia_cl_set_workspace_addr	21
3.2.27 xmedia_cl_get_workspace_addr	22
3.2.28 xmedia_cl_graph_get_private_data	22
3.2.29 xmedia_cl_graph_get_memory_reuse_type	23
3.2.30 xmedia_cl_graph_get_private_data_from_file	23
3.2.31 xmedia_cl_graph_get_private_data_from_buff	23
3.2.32 xmedia_cl_graph_query_model_info_from_file	24
3.2.33 xmedia_cl_graph_query_model_info_from_buff	24
3.2.34 xmedia_cl_graph_communicate	25
3.2.35 xmedia_cl_graph_decommunicate	25
3.2.36 xmedia_cl_graph_set_compress_config	25
3.2.37 xmedia_cl_graph_set_schedule_prio	26
3.2.38 运行层错误说明	26
4 驱动层用户使用流程	28
5 运行层用户使用流程	29

1 概述

1.1 软件栈

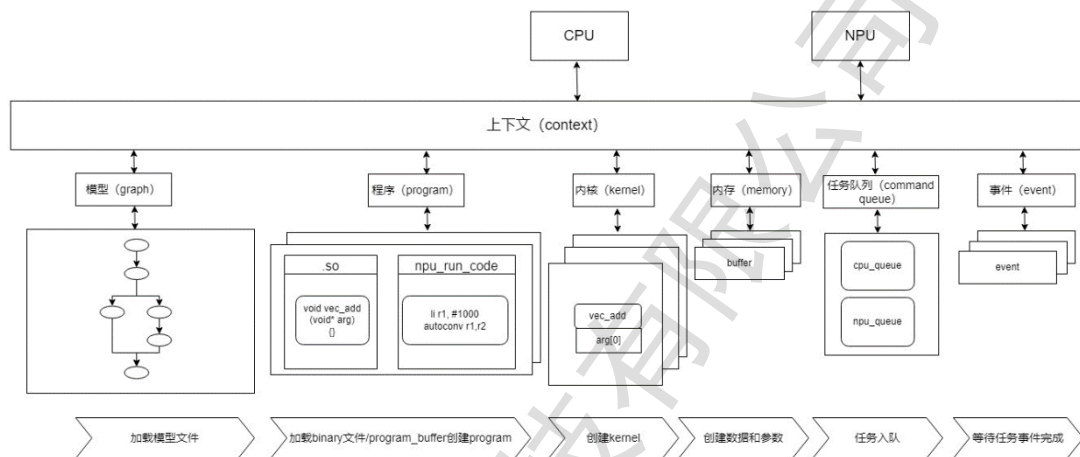


1.2 Feature List

1. 支持零拷贝模式，支持 ping-pong buffer
2. 支持同步、异步模式
3. 支持 JIT 和 AOT
4. 支持异构设备扩展
5. 内部无缓冲 buffer，无额外内存占用

1.3 基本介绍

XMEDIA_CL 是针对 NPU 设计的异构编程框架，主要是实现 CPU/NPU/DSP 等设备的统一管理和任务并行调度。



如图所示，上下文负责统一的设备资源管理，图示中分别含有 1 个 CPU 和 1 个 NPU 设备。然后创建两个运行队列。用户先加载模型文件。加载文件过程中 XMEDIA_CL 会根据 graph 结构自动创建程序对象和内核函数。创建完成后，XMEDIA_CL 再将内核函数放入到任务队列中，生成任务事件。

kernel 函数执行均采用异步模式，用户向任务队列提交命令后可执行其它工作而无需等待命令完成，如有必要等到命令完成，则可以通过事件显式地建立这个约束。

2 UMD API 参考

应用程序根据调用流程，通过给定的函数接口调用 NPU 驱动处理数据。

2.1 函数参数说明

xmedia_npu_proc_t

如图所示，结构体 xmedia_npu_proc_t 封装了 npu proc buff 的信息。

```
typedef struct {  
    xmedia_addr phy_addr;    // npu proc buff的物理地址  
    xmedia_u32 buff_len;    // npu proc buff的长度  
} xmedia_npu_proc_t;
```

2.2 API 接口

2.2.1 xmedia_npu_get_usage_rate

函数原型	xmedia_s32 xmedia_npu_get_usage_rate(xmedia_float *usage)
函数说明	获取 npu 利用率
参 数	*usage 表示 npu 利用率（输出）
返 回 值	返回值为 0 表示成功获取，其余值表示失败
约束说明	无

2.2.2 xmedia_npu_get_proc_info

函数原型	xmedia_s32 xmedia_npu_get_proc_info(xmedia_npu_proc_t *proc)
函数说明	获取 npu proc buff 的物理地址和长度
参 数	*proc 表示 npu proc buff 的物理地址和长度（输入）

返回值	返回值为 0 表示成功获取，其余值表示失败
约束说明	提供给 aisdk 使用

2.2.3 xmedia_npu_set_quick_start_flag

函数原型	xmedia_s32 xmedia_npu_set_quick_start_flag(xmedia_u32 flag)
函数说明	设置快启标志
参 数	flag 表示要设置的快启标志（输入），0 表示关闭快启（默认）
返回值	返回值为 0 表示成功获取，其余值表示失败
约束说明	快启设置使能，可以加速模型加载

2.2.4 xmedia_npu_get_quick_start_flag

函数原型	xmedia_s32 xmedia_npu_get_quick_start_flag(xmedia_u32 *flag)
函数说明	获取快启标志
参 数	*flag 表示获取快启标志（输出）
返回值	返回值为 0 表示成功获取，其余值表示失败
约束说明	无

3 运行层 API 参考

上层应用应该按照使用说明，通过给定函数接口调用 XMEDIA_CL API 下发任务和处理数据。

3.1 名词解释

- 设备(device)
硬件计算单元。任务队列将命令排队到设备上执行，XMEDIA_CL 中的设备主要是指 NPU/CPU/DSP 等。
- 上下文(context)
资源管理者。负责管理各个设备、设备可访问的内存、每个设备对应的任务队列、程序和各个内核函数。
- 模型图文件(graph)
编译器编译 AI 模型生成的二进制文件。支持异构指令。

3.2 运行时 API

3.2.1 函数参数说明

- xmedia_cl_status_e
枚举量，用于表示任务的各种状态，封装在事件中。
XMEDIA_CL_QUEUED：表示该任务已经进入任务队列中
XMEDIA_CL_SUBMITTED：表示该任务已经提交给设备
XMEDIA_CL_RUNNING：表示该任务在设备上正在运行
XMEDIA_CL_COMPLETED：表示该任务正常完成
XMEDIA_CL_FAILED：表示该任务执行失败

xmedia_cl_status_e
XMEDIA_CL_QUEUED
XMEDIA_CL_SUBMITTED
XMEDIA_CL_RUNNING
XMEDIA_CL_COMPLETED
XMEDIA_CL_FAILED

- xmedia_cl_device_type_e
枚举量，XMEDIA_CL_DEVICE_CPU 表示为 CPU 设备，XMEDIA_CL_DEVICE_NPU 表示为 NPU 设备，XMEDIA_CL_DEVICE_ALL 表示所有设备。

xmedia_cl_device_type_e
XMEDIA_CL_DEVICE_CPU
XMEDIA_CL_DEVICE_NPU
XMEDIA_CL_DEVICE_ALL

- xmedia_cl_data_type
枚举量，表示 tensor 的数据类型。

xmedia_cl_data_type
XMEDIA_CL_INT8
XMEDIA_CL_UINT8
XMEDIA_CL_INT16
XMEDIA_CL_UINT16
XMEDIA_CL_FP16
XMEDIA_CL_INT32
XMEDIA_CL_FP32
XMEDIA_CL_INT4
XMEDIA_CL_UINT12
XMEDIA_CL_UINT10
XMEDIA_CL_DATA_TYPE_END

- xmedia_cl_tensor_shape
结构体，ndims 表示 tensor 的维度数量，dims 表示 tensor 的维度信息，pch 表示 tensor 的 pitch 信息，xmedia_cl_data_type 表示 tensor 的数据类型。

xmedia_cl_tensor_shape
xmedia_cl_u32 ndims
xmedia_cl_u32 dims[XMEDIA_CL_MAX_DIMS_NUM]
xmedia_cl_u32 patch[XMEDIA_CL_MAX_DIMS_NUM]
xmedia_cl_data_type type

- xmedia_cl_tensor_quant
结构体，表示 tensor 的量化信息。

xmedia_cl_tensor_quant
xmedia_cl_float scale
xmedia_cl_s32 zp

- xmedia_cl_tensor
结构体，表示 tensor 的信息，tensor_id 表示 tensor 的标识，addr 表示 tensor 在内存中的地址，shape 表示 tensor 的形状，quant 表示 tensor 的量化信息，size 表示 tensor 的总大小，name 表示当前 tensor 的名字。

xmedia_cl_tensor
xmedia_cl_u32 tensor_id
void * addr
xmedia_cl_tensor_shape shape
xmedia_cl_tensor_quant quant
xmedia_cl_u32 size
xmedia_cl_s8 *name

- xmedia_cl_tensor_batch
结构体，表示输入 tensor 支持的 batch 信息，batch_count 表示支持的 batch 信息数量，batch 表示 tensor 的 batch 信息。

xmedia_cl_tensor_batch
xmedia_cl_u32 batch_count;
xmedia_cl_u32 batch[XMEDIA_CL_BATCH_NUM]

- xmedia_cl_tensor_info_inout

结构体，表示输入或者输出 tensor 的信息，num 表示数量，tensor 表示 tensor xmedia_cl_tensor

的信息，tensor_batch 表示 tensor 支持的 batch 信息，current_batch 表示当前 tensor 的 batch 信息。

xmedia_cl_tensor_info_inout
xmedia_cl_u32 num xmedia_cl_tensor *tensor xmedia_cl_tensor_batch* tensor_batch xmedia_cl_u32 *current_batch

- xmedia_cl_mem_info

结构体，表示模型的相关内存信息，worksize 表示模型 workspace 大小，weightsize 表示模型权重参数大小，inputsize 表示模型输入大小，outputsize 表示模型输出大小，codesize 表示模型 code 大小，memory_reuse_type 表示内存复用类型，private_data_size 表示 private_data 大小。

xmedia_cl_mem_info
xmedia_cl_u32 worksize xmedia_cl_u32 weightsize xmedia_cl_u32 inputsize xmedia_cl_u32 outputsize xmedia_cl_u32 codesize xmedia_cl_u32 memory_reuse_type xmedia_cl_u32 private_data_size

- xmedia_cl_model_info_type

枚举量，表示查询模型的信息类别。XMEDIA_CL_MEM_INFO 表示查询模型内存信息。

xmedia_cl_model_info_type
XMEDIA_CL_MEM_INFO

- xmedia_cl_profiling_type_e

枚举量，表示 profiling 工具的类型，XMEDIA_CL_PROFILING_TYPE_NORMAL 表示输出 npu_profiling_actual.txt 为常规模式（包含网络总 cycle 数，并行次数，TN 名称，每个 TN 开始及结束时当前 cycle 数，单个 TN 消耗的 cycle 数在整个网络的占比以及单个 TN 中具体的流水信息），

XMEDIA_CL_PROFILING_TYPE_SIMPLE 表示输出 npu_profiling_actual.txt 为简单模式（不显示单个 TN 中具体的流水信息）。

xmedia_cl_profiling_type_e
XMEDIA_CL_PROFILING_TYPE_NORMAL XMEDIA_CL_PROFILING_TYPE_SIMPLE

- xmedia_cl_profiling_flag_e
枚举量，表示 profiling 工具是开启或关闭状态，XMEDIA_CL_PROFILING_OFF 表示关闭，XMEDIA_CL_PROFILING_ON 表示开启。

xmedia_cl_profiling_flag_e
XMEDIA_CL_PROFILING_OFF XMEDIA_CL_PROFILING_ON

- xmedia_cl_profiling_params
结构体，表示 profiling 工具需要的参数。

xmedia_cl_profiling_params
xmedia_cl_profiling_type_e profiling_type xmedia_cl_profiling_flag_e profiling_flag

- xmedia_cl_memory_reuse_type
枚举量，表示内存复用的类型，XMEDIA_CL_WORKSPACE 表示内存不复用，XMEDIA_CL_WORKSPACE_INPUT 表示 workspace 与 input 复用，XMEDIA_CL_WORKSPACE_OUTPUT 表示 workspace 与 output 复用，XMEDIA_CL_WORKSPACE_INPUT_OUTPUT 表示 workspace，input，output 三者复用。

xmedia_cl_memory_reuse_type
XMEDIA_CL_WORKSPACE XMEDIA_CL_WORKSPACE_INPUT XMEDIA_CL_WORKSPACE_OUTPUT XMEDIA_CL_WORKSPACE_INPUT_OUTPUT

3.2.2 xmedia_cl_init

函数原型	xmedia_cl_s32 xmedia_cl_init()
函数说明	全局初始化，用于申请设备内存
参 数	无
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_OUT_OF_HOST_MEMORY，表示 npu 申请 uncached 内存失败
约束说明	无

3.2.3 xmedia_cl_uninit

函数原型	xmedia_cl_s32 xmedia_cl_uninit()
函数说明	全局释放，用于释放设备内存
参 数	无
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_UNINIT，表示未执行全局初 始化 失败返回错误码 XMEDIA_CL_INVALID_OPERATION，表示有资源 未释放
约束说明	无

3.2.4 xmedia_cl_get_device_ids

函数原型	xmedia_cl_s32 xmedia_cl_get_device_ids(xmedia_cl_device_type_e device_type, xmedia_cl_device_id *devices, xmedia_cl_u32 *num_devices)
函数说明	获取设备相关信息
参 数	设备类型，设备列表，设备数目
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_DEVICE_TYPE，表示传入 的非法的设备类型

	失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 num_devices 为空指针 失败返回错误码 XMEDIA_CL_INVALID_DEVICE，表示 devices 列表的数量小于 num_devices
约束说明	设备类型只允许 CPU/NPU/ALL 传入，其他类型报错 首次调用此接口前，num_devices 必须初始化为 0

3.2.5 xmedia_cl_release_device_ids

函数原型	xmedia_cl_s32 xmedia_cl_release_device_ids (xmedia_cl_device_id *devices, xmedia_cl_u32 *num_devices)
函数说明	释放设备相关信息
参 数	设备列表，设备数目
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_DEVICE，表示 devices 或 num_devices 为空指针
约束说明	调接口之前，应保证对应的 context 已释放完成

3.2.6 xmedia_cl_create_context

函数原型	xmedia_cl_context xmedia_cl_create_context(xmedia_cl_u32 num_devices, xmedia_cl_device_id *devices, xmedia_cl_s32 *err_code)
函数说明	创建 xmedia_cl_context 对象
参 数	需要创建 context 的设备数目，对应设备数目的编号列表，错误编号指针
返 回 值	创建好的 xmedia_cl_context 对象 如果创建失败，返回值为空指针 错误码 XMEDIA_CL_INVALID_VALUE 表示传入的 num_devices 值错误或 devices 为空指针 错误码 XMEDIA_CL_INVALID_DEVICE 表示传入的 devices 列表中为空设备
约束说明	1. 若设备列表为空，则创建失败 2. num_devices 应和设备列表的数目一致

3.2.7 xmedia_cl_release_context

函数原型	xmedia_cl_s32 xmedia_cl_release_context(xmedia_cl_context context)
函数说明	销毁 context 对象
参 数	待销毁的 xmedia_cl_context 对象
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_INVALID_CONTEXT, 表示 context 为空指针
约束说明	调该接口之前, 应保证对应的 command_queue, program 均已释放完成

3.2.8 xmedia_cl_graph_querysize_from_file

函数原型	xmedia_cl_s32 xmedia_cl_graph_querysize_from_file (const char *model, xmedia_cl_u32 *worksize, xmedia_cl_u32 *weightsize)
函数说明	从网络模型（文件）查询 workspace 与 weight 的大小
参 数	model 表示需要查询的网络模型的绝对路径（输入），worksize 表示查询到的 workspace 的大小（输出），weightsize 表示查询到的 weight 的大小（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_MODEL, 表示使用的模型无效或者被损坏 失败返回错误码 XMEDIA_CL_READ_MODEL_FAIL, 表示读取模型失败 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示无效的参数 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE, 表示使用错误类型的模型
约束说明	只有多分区模型才可以查询 workspace 与 weight

3.2.9 xmedia_cl_graph_querysize_from_buff

函数原型	xmedia_cl_s32 xmedia_cl_graph_querysize_from_buff (const char *model, xmedia_cl_u32 *worksize, xmedia_cl_u32 *weightsize)
------	---

函数说明	从网络模型（内存）查询 workspace 与 weight 的大小
参 数	model 表示需要加载的网络模型的内存地址（输入），worksize 表示查询到的 workspace 的大小（输出），weightsize 表示查询到的 weight 的大小（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_MODEL，表示使用的模型无效或者被损坏 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE，表示使用错误类型的模型
约束说明	只有多分区模型才可以查询 workspace 与 weight

3.2.10 xmedia_cl_graph_loadmodel_from_file

函数原型	xmedia_cl_s32 xmedia_cl_graph_loadmodel_from_file(xmedia_cl_context *context, const char *model, xmedia_cl_graph *graph)
函数说明	从文件加载解析网络模型
参 数	context 表示运行库的 context 对象（输入），model 表示需要加载的网络模型的绝对路径（输入），graph 表示返回的模型句柄（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_MODEL，表示使用的模型无效或者被损坏 失败返回错误码 XMEDIA_CL_READ_MODEL_FAIL，表示读取模型失败 失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR，表示内部函数调用失败 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE，表示使用错误类型的模型 失败返回错误码 XMEDIA_CL_ERROR_ADDR_ALIGN，表示地址未按要求对齐

约束说明	无
------	---

3.2.11 xmedia_cl_graph_loadmodel_from_buff

函数原型	xmedia_cl_s32 xmedia_cl_graph_loadmodel_from_buff (xmedia_cl_context *context, const char *model, xmedia_cl_graph *graph)
函数说明	从内存加载解析网络模型
参 数	context 表示运行库的 context 对象（输入），model 表示需要加载的网络模型的内存地址（输入），graph 表示返回的模型句柄（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_MODEL，表示使用的模型无效或者被损坏 失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR，表示内部函数调用失败 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE，表示使用错误类型的模型 失败返回错误码 XMEDIA_CL_ERROR_ADDR_ALIGN，表示地址未按要求对齐
约束说明	无

3.2.12 xmedia_cl_graph_loadmodel_from_file_withmem

函数原型	xmedia_cl_s32 xmedia_cl_graph_loadmodel_from_file_withmem(xmedia_cl_context *context, const char *model, void *workspace, xmedia_cl_u32 worksize, void *weight, xmedia_cl_u32 weightsize, xmedia_cl_graph *graph)
函数说明	从文件加载解析网络模型
参 数	context 表示运行库的 context 对象（输入），model 表示需要加载的网络模型的绝对路径（输入），workspace 表示 workspace 的地址（输入），worksize 表示 workspace 的大小（输入），weight 表示 weight 的地址（输入），weightsize 表示 weight 的大小（输入），graph 表示返回的模型句柄（输出）

返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_MODEL，表示使用的模型无效或者被损坏 失败返回错误码 XMEDIA_CL_READ_MODEL_FAIL，表示读取模型失败 失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR，表示内部函数调用失败 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE，表示使用错误类型的模型 失败返回错误码 XMEDIA_CL_ERROR_ADDR_ALIGN，表示地址未按要求对齐
约束说明	workspace 和 weight 的物理地址必须 16 字节（XM7606V13）或 8 字节（XM7206V11A）对齐且是 MMZ 内存，worksize 和 weightsize 的大小要大于等于通过 xmedia_cl_graph_querysize_from_file 获取到的大小。

3.2.13 xmedia_cl_graph_loadmodel_from_buff_withmem

函数原型	<pre>xmedia_cl_s32 xmedia_cl_graph_loadmodel_from_buff_withmem(xmedia_cl_context *context, const char *model, void *workspace, xmedia_cl_u32 worksize, void *weight, xmedia_cl_u32 weightsize, xmedia_cl_graph *graph)</pre>
函数说明	从内存加载解析网络模型
参 数	context 表示运行库的 context 对象（输入），model 表示需要加载的网络模型的内存地址（输入），workspace 表示 workspace 的地址（输入），worksize 表示 workspace 的大小（输入），weight 表示 weight 的地址（输入），weightsize 表示 weight 的大小（输入），graph 表示返回的模型句柄（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_MODEL，表示使用的模型无效或者被损坏 失败返回错误码 XMEDIA_CL_READ_MODEL_FAIL，表示读取模型失败

	失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR, 表示内部函数调用失败 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示无效的参数 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE, 表示使用错误类型的模型 失败返回错误码 XMEDIA_CL_ERROR_ADDR_ALIGN, 表示地址未按要求对齐
约束说明	workspace 和 weight 的物理地址必须 16 字节 (XM7606V13) 或 8 字节 (XM7206V11A) 对齐且是 MMZ 内存, worksizes 和 weightsizes 的大小要大于等于通过 xmedia_cl_graph_querysize_from_buff 获取到的大小。

3.2.14 xmedia_cl_graph_get_input

函数原型	xmedia_cl_s32 xmedia_cl_graph_get_input(xmedia_cl_graph graph, xmedia_cl_u32 input_num, xmedia_cl_tensor_info_inout *input)
函数说明	获取模型的输入 tensor 信息
参 数	graph 表示模型句柄 (输入), input_num 表示给定的输入数量 (输入), input 是返回的 tensor 信息 (输出)
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示无效的参数 失败返回错误码 XMEDIA_CL_INSUFFICIENT_SIZE, 表示给定的输入小于模型输入
约束说明	当 input_num 为 0 时, 需要调用两次该接口, 第一次获取模型输入数量, 再根据输入数量申请 input 的内存资源, 最后再调一次接口获取信息; 当 input_num 不为 0 时, 只需要调用一次该接口, input 内存资源需要提前根据 input_num 提前申请好, input_num 小于实际模型输入数量返回错误码 XMEDIA_CL_INSUFFICIENT_SIZE

3.2.15 xmedia_cl_graph_get_output

函数原型	xmedia_cl_s32 xmedia_cl_graph_get_output (xmedia_cl_graph graph, xmedia_cl_u32 output_num, xmedia_cl_tensor_info_inout *output)
函数说明	获取模型的输出 tensor 信息

参 数	graph 表示模型句柄（输入）， output_num 表示模型输出数量（输入）， output 是返回的 tensor 信息（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_INSUFFICIENT_SIZE，表示给定的输出小于模型输出
约束说明	当 output_num 为 0 时，需要调用两次该接口，第一次获取模型输出数量，再根据输出数量申请 output 的内存资源，最后再调一次接口获取信息；当 output_num 不为 0 时，只需要调用一次该接口，output 内存资源需要提前根据 output_num 申请好，output_num 小于实际模型输出数量返回错误码 XMEDIA_CL_INSUFFICIENT_SIZE

3.2.16 xmedia_cl_graph_set_inout

函数原型	xmedia_cl_s32 xmedia_cl_graph_set_inout(xmedia_cl_graph graph, const xmedia_cl_tensor_info_inout *input, const xmedia_cl_tensor_info_inout *output)
函数说明	设置模型的输入地址、输出地址
参 数	graph 表示模型句柄（输入）， input 表示输入的 tensor 信息（输入）， output 表示输出的 tensor 信息（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_UNSUPPORTED_DYNAMIC_OUTPUT，表示该模型不支持改变输出地址
约束说明	Input、output 必须为通过 xmedia_cl_graph_get_input、xmedia_cl_graph_get_output 获取到的 tensor，在动态设置输入或者输出地址时，输入或者输出的物理地址必须连续且是 MMZ 内存，而且每个输入或者输出的首地址必须 16 字节（XM7606V13）或 8 字节（XM7206V11A）对齐。

3.2.17 xmedia_cl_graph_process

函数原型	xmedia_cl_s32 xmedia_cl_graph_process(xmedia_cl_graph graph)
函数说明	同步接口，运行模型

参 数	graph 表示模型句柄（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR，表示内部函数调用失败
约束说明	无

3.2.18 xmedia_cl_graph_submit

函数原型	xmedia_cl_s32 xmedia_cl_graph_submit(xmedia_cl_graph graph, xmedia_cl_event *event)
函数说明	异步接口，运行模型
参 数	graph 表示模型句柄（输入），event 表示对应该模型的事件（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR，表示内部函数调用失败
约束说明	调用该函数后，需要根据返回的 event 查询模型有没有运行完成

3.2.19 xmedia_cl_graph_unload

函数原型	xmedia_cl_s32 xmedia_cl_graph_unload(xmedia_cl_graph graph)
函数说明	卸载网络模型，释放资源
参 数	graph 表示模型句柄（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_CALL_INTERFUNC_ERR，表示内部函数调用失败
约束说明	无

3.2.20 xmedia_cl_wait_for_events

函数原型	xmedia_cl_s32 xmedia_cl_wait_for_events(xmedia_cl_u32 events_num, xmedia_cl_event *events)
函数说明	等待 event 对应的任务执行完成
参 数	event 的数量, event 的列表
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示 num_events 为 0 或 events 为空指针 失败返回错误码 XMEDIA_CL_INVALID_EVENT_LIST, 表示 events 列表中的事件为空指针 失败返回错误码 XMEDIA_CL_WAIT_EVENT_FAILED, 表示列表中存在失败的事件, 需调用 xmedia_cl_query_event_status 查询
约束说明	等待事件的数目要和事件列表的大小一致, 否则报错

3.2.21 xmedia_cl_query_event_status

函数原型	xmedia_cl_s32 xmedia_cl_query_event_status(xmedia_cl_event event, xmedia_cl_u32 *status)
函数说明	查询 event 的状态
参 数	event 对象, 状态值指针
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_EVENT, 表示 event 为空指针 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示 status 为空指针
约束说明	无

3.2.22 xmedia_cl_release_event

函数原型	xmedia_cl_s32 xmedia_cl_release_event(xmedia_cl_event event)
函数说明	销毁 event 对象
参 数	待销毁的 event 对象

返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_EVENT，表示 event 为空指针
约束说明	无

3.2.23 xmedia_cl_set_event_callback

函数原型	xmedia_cl_s32 xmedia_cl_set_event_callback(xmedia_cl_event event, xmedia_cl_s32 status, void(*notify)(xmedia_cl_event, xmedia_cl_s32, void*), void *data)
函数说明	设置 event 的回调函数
参 数	event 表示事件（输入），status 表示事件触发状态（输入）notify 为回调函数指针（输入），data 为回调函数参数（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示回调函数为空或触发状态不对 失败返回错误码 XMEDIA_CL_INVALID_EVENT，表示事件为空指针
约束说明	无

3.2.24 xmedia_cl_graph_set_dynamic_batch_size

函数原型	xmedia_cl_s32 xmedia_cl_graph_set_dynamic_batch_size(xmedia_cl_graph graph, xmedia_cl_tensor_info_inout *input, xmedia_cl_u32 index, xmedia_cl_u32 batch_size)
函数说明	设置模型输入 tensor 的 batch 的值
参 数	graph 表示模型句柄（输入），input 表示输入 tensor 信息（输入），index 表示第几个输入 tensor（输入），batch_size 表示需要设置的 batch 数量（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 或 input 为空指针或 index 传入非法值或 batch_size 为 0 失败返回错误码 XMEDIA_CL_OUT_OF_MAX_BATCH，表示设置的 batch_num 超出最大范围

约束说明	不支持与内存复用功能同时使用
------	----------------

3.2.25 xmedia_cl_set_profiling

函数原型	xmedia_cl_s32 xmedia_cl_set_profiling(xmedia_cl_graph graph,xmedia_cl_profiling_params profiling_params)
函数说明	设置 profiling 是否开启及类型
参 数	graph 表示模型句柄（输入）， profiling_params 表示需要设置的参数，包含类型及是否开启 profiling 功能（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 为空指针 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 profiling_params 中 profiling_type 不为 XMEDIA_CL_PROFILING_TYPE_NORMAL 或 XMEDIA_CL_PROFILING_TYPE_SIMPLE 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 profiling_params 中 profiling_flag 不为 XMEDIA_CL_PROFILING_ON 或 XMEDIA_CL_PROFILING_OFF 失败返回错误码 XMEDIA_CL_NOT_FIND_FILE，表示找不到 cycle_bin_file.bin 文件
约束说明	只能在单进程单线程单 batch 模式下使用

3.2.26 xmedia_cl_set_workspace_addr

函数原型	xmedia_cl_s32 xmedia_cl_set_workspace_addr(xmedia_cl_context context, void *workspace_addr, xmedia_cl_u32 workszize)
函数说明	设置 workspace 共享内存的首地址和大小到 context
参 数	context 表示运行库的 context 对象（输入）， workspace 表示共享内存的首地址（输入）， workszize 表示共享内存的大小（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 为空指针或者 workspace_addr 为空指针或者 workszize 为 0 失败返回错误码 XMEDIA_CL_CONTEXT_WORKSPACE_FAIL，表

	示 workspace 共享内存已被设置
约束说明	不支持与切分后的子网络模型共用 workspace 内存 不支持与内存复用功能同时使用 非 workspace 模式下，不能与多模型内存复用功能共用 异构模型不支持 workspace 内存复用

3.2.27 xmedia_cl_get_workspace_addr

函数原型	xmedia_cl_s32 xmedia_cl_get_workspace_addr(xmedia_cl_context context, void **workspace_addr, xmedia_cl_u32 *workspace)
函数说明	获取 workspace 共享内存的首地址和大小
参 数	context 表示运行库的 context 对象（输入），workspace_addr 表示从 context 获取的 workspace 共享内存首地址（输出），workspace 表示从 context 获取的 workspace 共享内存大小（输出）。
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 为空指针或者 workspace_addr 为空指针或者 workspace 为 0 或者共享内存地址为空指针或者共享内存大小为 0
约束说明	无

3.2.28 xmedia_cl_graph_get_private_data

函数原型	xmedia_cl_s32 xmedia_cl_graph_get_private_data(xmedia_cl_graph graph, void **private_data, xmedia_cl_u32 *data_size)
函数说明	获取 private_data 的首地址和大小
参 数	graph 表示模型句柄（输入），private_data 表示获取的 private_data 的首地址（输出），data_size 表示获取的 private_data 的大小（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 为空指针或者 private_data 为空指针或者 data_size 为空指针 失败返回错误码 XMEDIA_CL_PRIVATE_DATA_NULL，表示模型中的 private_data 为空指针或 data_size 为 0

约束说明	无
------	---

3.2.29 xmedia_cl_graph_get_memory_reuse_type

函数原型	xmedia_cl_s32 xmedia_cl_graph_get_memory_reuse_type(xmedia_cl_graph graph, xmedia_cl_memory_reuse_type *memory_type)
函数说明	获取内存复用类型
参 数	graph 表示模型句柄（输入），memory_type 表示获取的内存复用的类型（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 为空指针
约束说明	不支持与 xmedia_cl_set_workspace_addr 同时使用 不支持与设置动态 batch 功能同时使用

3.2.30 xmedia_cl_graph_get_private_data_from_file

函数原型	xmedia_cl_s32 xmedia_cl_graph_get_private_data_from_file(const char *model, void *private_data, xmedia_cl_u32 data_size)
函数说明	从网络模型（文件）获取 private_data
参 数	model 表示需要加载的网络模型文件路径（输入），private_data 表示获取的 private_data 的首地址（输入），data_size 表示获取的 private_data 的大小（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 model 为空指针或者 private_data 为空指针 失败返回错误码 XMEDIA_CL_PRIVATE_DATA_SIZE_ERROR，表示输入的 data_size 小于从文件获取到的 private_data 的大小
约束说明	无

3.2.31 xmedia_cl_graph_get_private_data_from_buff

函数原型	xmedia_cl_s32 xmedia_cl_graph_get_private_data_from_buff(const char *model, void *private_data, xmedia_cl_u32 data_size)
------	---

函数说明	从网络模型（内存）获取 private_data
参 数	model 表示需要加载的网络模型内存地址（输入），private_data 表示获取的 private_data 的首地址（输入），data_size 表示获取的 private_data 的大小（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 model 为空指针或者 private_data 为空指针 失败返回错误码 XMEDIA_CL_PRIVATE_DATA_SIZE_ERROR，表示输入的 data_size 小于从内存获取到的 private_data 的大小
约束说明	无

3.2.32 xmedia_cl_graph_query_model_info_from_file

函数原型	xmedia_cl_s32 xmedia_cl_graph_query_model_info_from_file(const char *model, void *info, xmedia_cl_model_info_type info_type)
函数说明	从网络模型（文件）查询模型信息
参 数	model 表示需要加载的网络模型的文件路径（输入），info 表示查询到的模型信息（输出），info_type 表示查询的信息类型（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数 失败返回错误码 XMEDIA_CL_UNSUPPORTED_INFO_TYPE，表示不支持查询此类型信息
约束说明	目前只支持查询 XMEDIA_CL_MEM_INFO 类型信息

3.2.33 xmedia_cl_graph_query_model_info_from_buff

函数原型	xmedia_cl_s32 xmedia_cl_graph_query_model_info_from_buff(const char *model, void *info, xmedia_cl_model_info_type info_type)
函数说明	从网络模型（内存）查询模型信息
参 数	model 表示需要加载的网络模型的内存地址（输入），info 表示查询到的模型信息（输出），info_type 表示查询的信息类型（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示无效的参数

	失败返回错误码 XMEDIA_CL_UNSUPPORT_INFO_TYPE, 表示不支持查询此类型信息
约束说明	目前只支持查询 XMEDIA_CL_MEM_INFO 类型信息

3.2.34 xmedia_cl_graph_communicate

函数原型	xmedia_cl_s32 xmedia_cl_graph_communicate(xmedia_cl_graph graph, xmedia_cl_u32 *model_id)
函数说明	获取 aibnr 模型的 model_id
参 数	graph 表示模型句柄（输入），model_id 表示 aibnr 模型的 id（输出）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示 graph 为空指针或 model_id 为空指针 失败返回错误码 XMEDIA_CL_ERROR_MODEL_TYPE, 表示模型类型不匹配
约束说明	model_id 是内核态句柄，只针对 aibnr 模型使用

3.2.35 xmedia_cl_graph_decommunicate

函数原型	xmedia_cl_s32 xmedia_cl_graph_decommunicate (xmedia_cl_graph graph, xmedia_cl_u32 model_id)
函数说明	销毁 aibnr 模型的 model_id
参 数	graph 表示模型句柄（输入），model_id 表示 aibnr 模型的 id（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE, 表示 graph 为空指针
约束说明	model_id 是内核态句柄，只针对 aibnr 模型使用

3.2.36 xmedia_cl_graph_set_compress_config

函数原型	xmedia_cl_s32 xmedia_cl_graph_set_compress_config (xmedia_cl_graph graph, void *device_head_addr, xmedia_cl_u32 head_size, xmedia_cl_u32 break_line)
函数说明	设置模型数据压缩

参 数	graph 表示模型句柄（输入）， device_head_addr 表示压缩 head 信息的地址（输入）， head_size 表示 head 信息的大小（输入）， break_line 表示分片阈值（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示输入参数无效 失败返回错误码 XMEDIA_CL_INVALID_OPERATION，表示不支持压缩操作
约束说明	XM7606V13、XM7206V11a 暂不支持

3.2.37 xmedia_cl_graph_set_schedule_prio

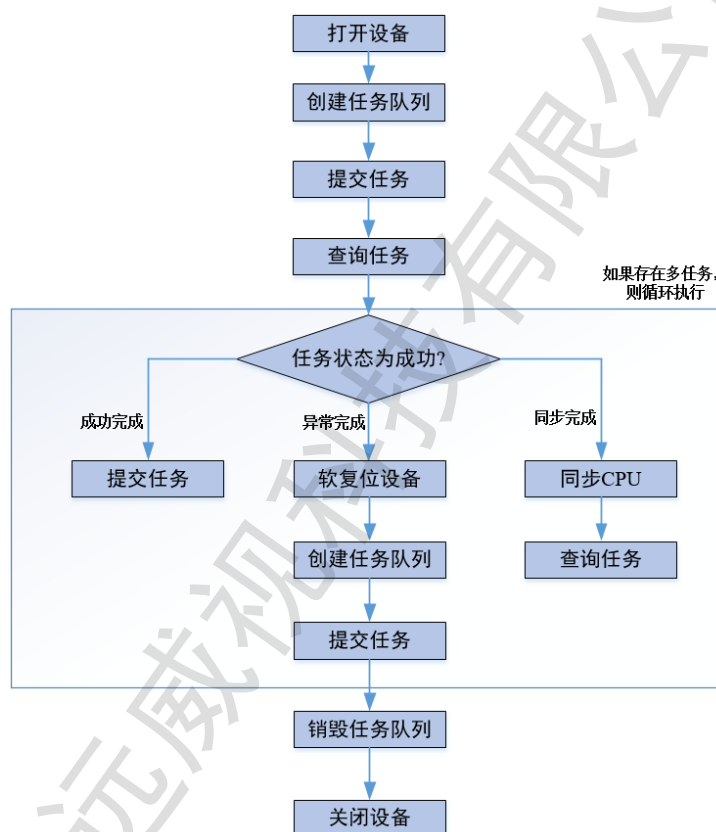
函数原型	xmedia_cl_s32 xmedia_cl_graph_set_schedule_prio(xmedia_cl_graph graph, xmedia_cl_u32 prio)
函数说明	设置任务优先级
参 数	graph 表示模型句柄（输入）， prio 表示需要设置的任务优先级（输入）
返 回 值	成功返回 XMEDIA_CL_SUCCESS 失败返回错误码 XMEDIA_CL_INVALID_VALUE，表示 graph 为空指针或 prio 超过可设置优先等级
约束说明	目前支持设置四个任务优先级，从 0~3 优先级依次升高

3.2.38 运行层错误说明

错误值	错误码	解释
0	XMEDIA_CL_SUCCESS	运行成功
-6	XMEDIA_CL_OUT_OF_HOST_MEMORY	内部申请内存失败
-30	XMEDIA_CL_INVALID_VALUE	非法的输入参数
-31	XMEDIA_CL_INVALID_DEVICE_TYPE	非法的设备类型
-33	XMEDIA_CL_INVALID_DEVICE	非法的设备句柄
-34	XMEDIA_CL_INVALID_CONTEXT	非法的上下文句柄

-56	XMEDIA_CL_WAIT_EVENT_FAILED	等待的事件失败
-57	XMEDIA_CL_INVALID_EVENT_LIST	非法的事件列表
-58	XMEDIA_CL_INVALID_EVENT	非法的事件句柄
-59	XMEDIA_CL_INVALID_OPERATION	非法操作资源未释放
-60	XMEDIA_CL_INVALID_UNINIT	未执行全局初始化
-61	XMEDIA_CL_INVALID_BUFFER_SIZE	非法申请 NPU 内存
-63	XMEDIA_CL_ALREADY_INIT	已完成全局初始化
-64	XMEDIA_CL_INVALID_MODEL	无效的模型
-65	XMEDIA_CL_READ_MODEL_FAIL	读模型失败
-66	XMEDIA_CL_INSUFFICIENT_SIZE	给定的数量小于实际数量
-67	XMEDIA_CL_CALL_INTERFUNC_ERR	内部函数调用失败
-68	XMEDIA_CL_ERROR_MODEL_TYPE	使用不匹配类型的模型
-69	XMEDIA_CL_ERROR_ADDR_ALIGN	地址未按要求对齐
-70	XMEDIA_CL_UNSUPPORT_DYNAMIC_OUTPUT	不支持动态设置输出地址
-73	XMEDIA_CL_ERROR_PROC_TYPE	打印 PROC 信息失败
-74	XMEDIA_CL_OUT_OF_MAX_BATCH	超出最大 BATCH 数
-75	XMEDIA_CL_NOT_FIND_FILE	找不到 cycle_bin_file.bin 文件
-76	XMEDIA_CL_CONTEXT_WORKSPACE_FAIL	context 中的 workspace 已经被设置
-77	XMEDIA_CL_PRIVATE_DATA_NULL	模型中 PRIVATE_DATA 为空
-78	XMEDIA_CL_PRIVATE_DATA_SIZE_ERROR	PRIVATE_DATA_SIZE 错误
-79	XMEDIA_CL_UNSUPPORT_INFO_TYPE	不支持查询模型此类型信息

4 驱动层用户使用流程



上图是根据应用开发中的典型功能抽象出主要的接口同步调用流程，提交任务后需要查询任务状态，直到状态为完成，根据不同的任务状态决定继续提交任务，还是软复位设备或者发出同步要求，异步调用流程可以多次提交任务，分别查询任务状态。

5 运行层用户使用流程

支持异步调用。运行网络模型可以选择使用 `xmedia_cl_graph_process` 同步等到运行完成，也可以选择使用 `xmedia_cl_graph_submit` 提交运行，再使用 `xmedia_cl_query_event_status` 查询任务完成的状态。

支持用户设置 batch 值。batch 值默认为 1，当用户需要更改 batch 值为 n 时，需自行申请一块可以存放 n 个输入（输出）大小的空间（n 个输入（输出）地址须连续）并调用 `xmedia_cl_graph_set_dynamic_batch_size` 接口更改 batch 值，目前支持的最大 batch 数目为 32。

支持用户设置回调函数。用户在调用 `xmedia_cl_graph_submit` 后得到 event 句柄，调用 `xmedia_cl_set_event_callback` 设置回调函数，根据 event 的状态自动触发回调函数。

支持 profiling 功能查看单个网络的具体信息。用户可以在调用 `xmedia_cl_graph_set_inout` 前调用 `xmedia_cl_set_profiling` 函数开启 profiling 功能，在 `xmedia_cl_graph_process` 后关闭 profiling 功能。

支持模型内存复用，用户可以在编译模型时确定模型是否进行内存复用以及模型内存复用的类型，在运行层调用 `xmedia_cl_graph_get_memory_reuse_type` 函数获取模型内存复用的类型。

