Introduction to CNML&CNRT APIs

This document introduces some important CNML™/CNRT™ interfaces used in the design of CNML/CNRT integration. Before introducing CNML/CNRT interfaces, we will firstly introduce an enumeration type called cnmlStatus_t, which is used for returning the runtime status of CNML interfaces.

1. cnmlStatus_t

Introduction to cnmlStatus_t	The cnmlStatus_t is used for returning the runtime status of CNML interfaces.
Introduction to Enumerated Values of cnmlStatus_t	
Enumerated Values	**Meaning of the Enumerated Values**
CNML_STATUS_NODEVICE	The needed device cannot be accessed.
CNML_STATUS_SUCCESS	The interface succeeds.
CNML_STATUS_DOMAINERR	An argument is out of its mathematical domain.
CNML_STATUS_INVALIDARG	There is an invalid argument.
CNML_STATUS_LENGTHERR	A value exceeds its valid size.
CNML_STATUS_OUTOFRANGE	An index error happens in the interface.
CNML_STATUS_RANGEERR	The computation result of the interface is out of the valid range.
CNML_STATUS_OVERFLOWERR	The computation result of the interface overflows.
CNML_STATUS_UNDERFLOWERR	The computation result of the interface underflows.
CNML_STATUS_INVALIDPARAM	There is an invalid parameter.
CNML_STATUS_BADALLOC	MLU memory allocation fails.
CNML_STATUS_BADTYPEID	An invalid typeid, which means that the pointer “p” in the expression “typeid(*p)” is a null pointer.
CNML_STATUS_BADCAST	An error occurs when the interface is performing type cast.
CNML_STATUS_UNSUPPORT	CNML cannot support the operator on current device.

Next, we will introduce some important CNML interfaces used in the design of CNML/CNRT integration.

2. Interfaces about the CNML tensor descriptor

2.1 cnmlCreateCpuTensor

Introduction to the Interface	Create a cnmlCpuTensor as the CNML CPU tensor descriptor of a tensor and set attributes, such as shape, data type*, etc.*, for the cnmlCpuTensor.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
cpu_tensor	cnmlCpuTensor_t*	Point to a cnmlCpuTensor_t instance. If the interface succeeds, the cnmlCpuTensor_t instance which the "cpu_tensor" points to will point to the created cnmlCpuTensor instance.	cnmlCpuTensor_t is a pointer of a cnmlCpuTensor instance.
data_type	cnmlDataType_t	Data type of the tensor	None
data_order	cnmlDataOrder_t	Order of the dimensions, namely, n, c, h and w, of the tensor	None
n,c,h,w	int	Size of the dimensions, namely, n, c, h and w, of the tensor	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

2.2 cnmlDestroyCpuTensor

Introduction to the Interface	Destroy a cnmlCpuTensor instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**

cpu_tensor	cnmlCpuTensor_t*	Point to a cnmlCpuTensor_t instance. If the interface succeeds, the cnmlCpuTensor instance will be destroyed and the cnmlCpuTensor_t instance which the "cpu_tensor" points to will be updated to "NULL".	cnmlCpuTensor_t is a pointer of a cnmlCpuTensor instance.
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

2.3 cnmlCreateTensor

Introduction to the Interface	Create a cnmlTensor as the CNML MLU tensor descriptor of a tensor and set attributes, such as shape, data type, etc., for the cnmlTensor.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
tensor	cnmlTensor_t*	Point to a cnmlTensor_t instance. If the interface succeeds, the cnmlTensor_t instance which the "tensor" points to will point to the created cnmlTensor instance.	cnmlTensor_t is a pointer of a cnmlTensor instance.
tensor_type	cnmlTensorType_t	CNML tensor type of the tensor	None
data_type	cnmlDataType_t	Data type of the tensor.	None
n,c,h,w	int	Size of the dimensions, namely, n, c, h and w, of the tensor.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

2.4 cnmlDestroyTensor

Introduction to the Interface	Destroy a cnmlTensor instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
tensor	cnmlTensor_t*	Point to a cnmlTensor_t instance. If the interface succeeds, the cnmlTensor instance will be destroyed and the cnmlTensor_t instance which the "tensor" points to will be updated to "NULL".	cnmlTensor_t is a pointer of a cnmlTensor instance.
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

2.5 cnmlBindConstData

Introduction to the Interface	For a tenor, the interface binds its cnmlCpuTensor and data address to its cnmlTensor.		
Remark	Only CNML_FILTER and CNML_CONST tensors need to be bound to constant data.		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
tensor	cnmlTensor_t	Point to the cnmlTensor instance of a tensor to be bound to constant data.	None
cpu_tensor	cnmlCpuTensor_t	Point to the cnmlCpuTensor instance of the tensor to be bound to constant data.	None
cpu_tensor_ptr	void*	The data address at host end of the tensor to be bound to	None

		constant data	
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface.	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

3. Interfaces about MLU memory management

3.1 cnmlMallocBuffer

Introduction to the Interface	Allocate MLU memory for the tensor described by the parameter “tensor”.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
tensor	cnmlTensor_t	Point to a cnmlTensor instance describing the tensor for which MLU memory is allocated.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
void*	If the interface succeeds, it will return the pointer of the allocated MLU memory.	None	

3.2 cnmlFreeBuffer

Introduction the Interface	If the parameter “ptr” is not null and no one is using the MLU memory which “ptr” points to, the interface will free the MLU memory which the “ptr” points to.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
ptr	void*	Point to the MLU memory which needs to be freed.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

3.3 cnmlMemcpyTensorToDevice

Introduction the Interface	Copy data from host memory to MLU memory.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
cpu_tensor	cnmlCpuTensor_t	Point to a cnmlCpuTensor instance which is the CNML host descriptor of the source data.	None
src	void*	Point to the source data in host memory.	None
cnml_tensor	cnmlTensor_t	Point to a cnmlTensor instance which is the CNML MLU descriptor of the source data.	None
dst	void*	Point to the destination MLU memory.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

3.4 cnmlMemcpyTensorToHost

Introduction the Interface	Copy data from MLU memory to host memory.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
cnml_tensor	cnmlTensor_t	Point to a cnmlTensor instance which is the CNML MLU descriptor of the source data.	None
src	void*	Point to the source data in MLU memory.	None
cpu_tensor	cnmlCpuTensor_t	Point to a cnmlCpuTensor instance which is the CNML host descriptor	None

		of the source data.	
dst	void*	Point to the destination host memory.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4. Interfaces about the CNML BaseOp

Every CNML basic operator has its own creation and forward interfaces. This section will introduce some interfaces related to convolution and activation operators as examples, and also some basic interfaces about the BaseOp.

4.1 cnmlCreateConvOpParam

Introduction the Interface	Create a parameter structure for a CNML convolution basic operator.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
param	cnmlConvOpParam_t*	Point to a cnmlConvOpParam_t instance. If the interface succeeds, the cnmlConvOpParam_t instance will point to the created cnmlConvOpParam instance.	None
stride_height	int	Stride size in height direction	None
stride_width	int	Stride size in width direction	None
dilation_height	int	Dilation coefficient in height direction	None
dilation_width	int	Dilation coefficient in width direction	None
pad_height	int	Height of pad	None
pad_width	int	Width of pad	None
sparse	cnmlSparseMode_t	The convolution	CNML_NoSparse/

		operator is in sparse mode or not.	CNML_Sparse
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4.2 cnmlCreateConvOp

Introduction to the Interface	Create a CNML convolution basic operator.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t*	Point to a cnmlBaseOp_t instance. If the interface succeeds, the cnmlBaseOp_t instance will point to the created ConvOp instance.	None
param	cnmlConvOpParam_t	Point to a convolution parameter structure instance of a convolution operator.	None
inputTensor	cnmlTensor_t	Point to a cnmlTensor instance of the input tensor.	None
outputTensor	cnmlTensor_t	Point to a cnmlTensor instance of the output tensor.	None
filterTensor	cnmlTensor_t	Point to a cnmlTensor instance of the weight tensor.	None
biasTensor	cnmlTensor_t	Point to a cnmlTensor instance of the bias tensor.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4.3 cnmlCreateActiveOp

Introduction to the Interface	Create a CNML activation basic operator.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t*	Point to a cnmlBaseOp_t instance. If the interface succeeds, the cnmlBaseOp_t instance will point to the created ActiveOp instance.	None
function	cnmlActiveFunction_t	The activation function type of the ActiveOp	cnmlActiveFunction_t is an enumeration type.
input	cnmlTensor_t	Point to a cnmlTensor instance of the input tensor.	None
output	cnmlTensor_t	Point to a cnmlTensor instance of the output tensor.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4.4 cnmlComputeConvOpForward

Introduction to the Interface	Execute the forward computation of a ConvOp instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t	Point to a created and compiled ConvOp instance.	None
input	void*	Point to the MLU memory of input tensor data.	None

output	void*	Point to the MLU memory which is used for saving output tensor data.	None
compute_forw_param	cnrtInvokeFuncParam_t*	Point to a cnrtInvokeFuncParam instance.	None
queue	cnrtQueue_t	Point to a cnrtQueue instance.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4.5 cnmlComputeActiveOpForward

Introduction to the Interface	Execute the forward computation of an ActiveOp instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t	Point to a created and compiled ActiveOp instance.	None
input	void*	Point to the MLU memory of input tensor data.	None
output	void*	Point to the MLU memory which is used for saving output tensor data.	None
compute_forw_param	cnrtInvokeFuncParam_t*	Point to a cnrtInvokeFuncParam instance.	None
queue	cnrtQueue_t	Point to a cnrtQueue instance.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4.6 cnmlCompileBaseOp

Introduction to	Compile a created BaseOp instance.

the Interface			
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t	Point to a created BaseOp instance.	None
version	cnmlCoreVersion_t	Version of the MLU kernel	None
model_parallelism	int	The number of model parallelism	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

4.7 cnmlDestroyBaseOp

Introduction to the Interface	Destroy a created BaseOp instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t*	Point to a cnmlBaseOp_t instance which points to a created BaseOp instance. After destroying the BaseOp instance, the cnmlBaseOp_t instance will be updated to NULL.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

5. Interfaces about the CNML FusionOp

5.1 cnmlCreateFusionOp

Introduction to	Create a FusionOp instance.

the Interface			
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlFusionOp_t*	Point to a cnmlFusionOp_t instance. If the interface succeeds, the cnmlFusionOp_t instance will point to the created FusionOp instance.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

5.2 cnmlFuseOp

Introduction to the Interface	Add a BaseOp to a created FusionOp.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlBaseOp_t	Point to a created BaseOp instance which will be added to the FusionOp instance.	None
fusionOp	cnmlFusionOp_t	Point to the FusionOp instance which the BaseOp instance add to.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

5.3 cnmlSetFusionIO

Introduction to the Interface	Set I/O for a FusionOp instance.
Remark	None
Parameter Explanation	

Parameter Name	Data Type	Explanation	Remark
op	cnmlFusionOp_t	Point to a FusionOp instance which needs to be set I/O.	None
inputs	cnmlTensor_t*	Point to an array of cnmlTensor instances of input tensors.	None
inputNum	int	The number of cnmlTensor instances of input tensors	None
outputs	cnmlTensor_t*	Point to an array of cnmlTensor instances of output tensors.	None
outputNum	int	The number of cnmlTensor instances of output tensors	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

5.4 cnmlCompileFusionOp

Introduction to the Interface	Compile a created FusionOp instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlFusionOp_t	Point to a created FusionOp instance.	None
version	cnmlCoreVersion_t	Version of the MLU kernel	None
model_parallelism	int	The number of model parallelism	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

5.5 cnmlComputeFusionOpForward

Introduction to the Interface	Execute the forward computation of a FusionOp instance.
Remark	None

Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlFusionOp_t	Point to a created and compiled FusionOp instance.	None
inputs	void*	Point to an array of pointers of the MLU memory of input tensor data.	None
inputNum	int	The number of input tensors	None
outputs	void*	Point to an array of pointers of the MLU memory which is used for saving output tensor data.	None
outputNum	int	The number of output tensors	None
compute_forw_param	cnrtInvokeFuncParam_t*	Point to a cnrtInvokeFuncParam instance.	None
queue	cnrtQueue_t	Point to a cnrtQueue instance.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

5.6 cnmlDetroyFusionOp

Introduction to the Interface	Destroy a created FusionOp instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
op	cnmlFusionOp_t*	Point to a cnmlFusionOp_t instance which points to a created FusionOp instance. After destroying the FusionOp instance, the cnmlFusionOp_t instance will be	None

		updated to NULL.	
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnmlStatus_t	Runtime status of the CNML interface	See enumerated values of cnmlStatus_t in section cnmlStatus_t.	

6. Interfaces about the cnrtQueue

6.1 cnrtCreateQueue

Introduction to the Interface	Create a cnrtQueue instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
pQueue	cnrtQueue_t*	Point to a cnrtQueue_t instance. If the interface succeeds, the cnrtQueue_t instance will point to the created cnrtQueue instance.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnrtRet_t	Runtime status of the CNRT interface. If the interface succeeds, it will return to CNRT_RET_SUCCESS.	None	

6.2 cnrtSyncQueue

Introduction to the Interface	Push a synchronization task into a created cnrtQueue instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
queue	cnrtQueue_t	Point to a created cnrtQueue instance.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	

cnrtRet_t	Runtime status of the CNRT interface. If the interface succeeds, it will return to CNRT_RET_SUCCESS.	None

6.3 cnrtDestroyQueue

Introduction to the Interface	Destroy a created cnrtQueue instance.		
Remark	None		
Parameter Explanation			
Parameter Name	**Data Type**	**Explanation**	**Remark**
queue	cnrtQueue_t	Point to a created cnrtQueue instance which needs to be destroyed.	None
Return Value Explanation			
Data Type	**Explanation**	**Remark**	
cnrtRet_t	Runtime status of the CNRT interface. If the interface succeeds, it will return to CNRT_RET_SUCCESS.	None	