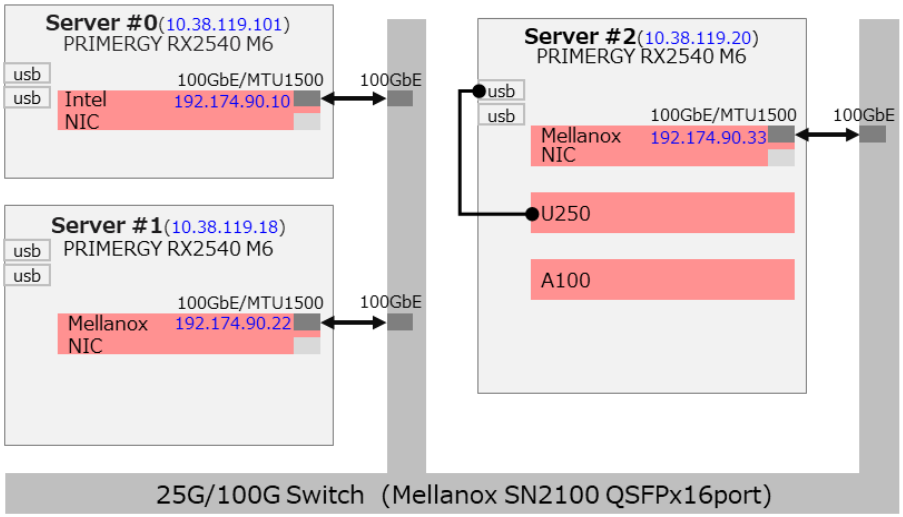
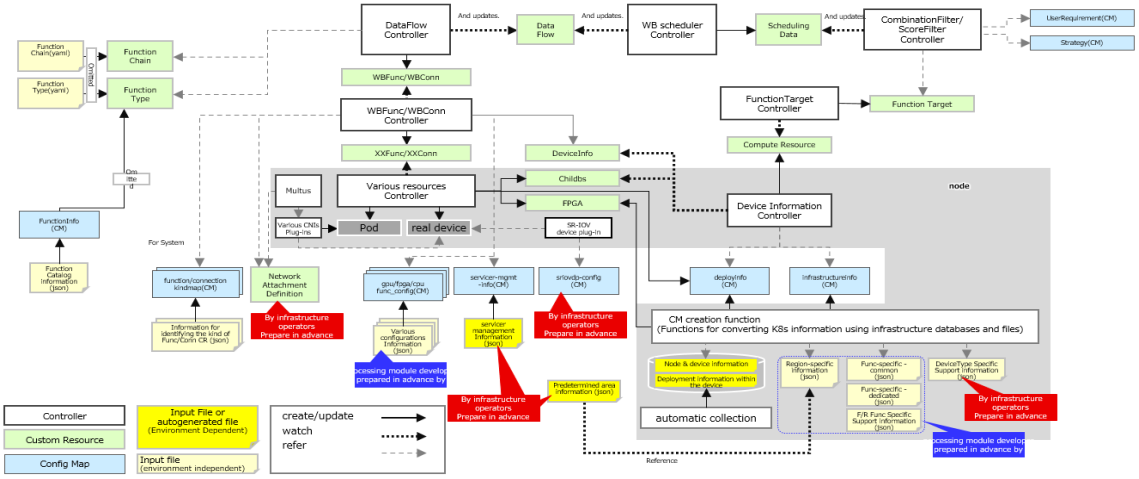
OpenKasugai-Controller
Install Manual(Attachment)

Sheet Name	Description
1.assumed environment diagram, etc.	Physical and software configurations assumed in the construction procedures
2.YAML about	Explanation of the settings Supplementary information sheet available
3.Description of the input data (JSON) used to create CM	Explanation of settings and values. See also supplementary sheet Supplementary information sheet available
4. CRC YAML for daemonset about	Explaining the Contents of CRC's YAML for Daemonset
5. Description of device_info.txt	automatic collection&CM Description of the output of the creation tool

v1.0.0



Physical configuration diagram



Overall software configuration diagram

We describe a YAML file you need to create to deploy DataFlow by using ~/controller/sample-data/sample-data-demo/yaml/dataflows/test-ext-1/df-test-ext-1-1 as an example.
For details on how to set the network information (IP address and port number) used by Pod of each processing module, refer to "2. (Supplement)"

DataFlow YAML	Description	Remarks
apiVersion: example.com/v1		
kind: DataFlow		
metadata:		
name: "df-test-3-1-1-1"	Be set by the user	
namespace: "test01"	Be set by the user	
spec:		
functionChainRef:		
name: "cpu-decode-cpu-filter-resize-2types-high-infe	FunctionChain metadata.Name used by DataFlow	
namespace: "chain-imgproc"	FunctionChain metadata.Namespace used by DataFlow	
requirements:	Describe the requirements that must be met during scheduling	Only function chain requirements can be specified in the current prototype (common requirements apply to all functions)
all:	Describe requirements for function chain as a whole	
capacity: 15	Describe the assumed load (fps)	Each connection and the estimated load for each connection (fps)
functionUserParameter:		
- functionKey: decode-main	CPU decoding Function identifier	
userParams:		
ipAddress: 192.174.90.101/24	The own IP address. Set as the IP address of Pod 2nd NIC	Specify the IP address of the same subnet as the physical IP address of the 100GNIC that created the VF of the SR-IOV.
inputPort: 5004	own port number	
outputIPAddress: 192.174.90.111	Destination (CPU filter/resize) IP address	
outputPort: 15000	Destination (CPU filter/resize) port number	
- functionKey: filter-resize-high-infer-main	CPU filter/resize Function identifier	
userParams:		
ipAddress: 192.174.90.111/24	The own IP address. Set as the IP address of Pod 2nd NIC	Specify the IP address of the same subnet as the physical IP address of the 100GNIC that created the VF of the SR-IOV.
inputPort: 15000	own port number	
outputIPAddress: 192.174.90.121	Destination (copy branch) IP address	
outputPort: 16000	Destination (copy branch) port number	
- functionKey: copy-branch-main	copy branch Function identifier	
userParams:		
ipAddress: 192.174.90.121/24	The own IP address. Set as the IP address of Pod 2nd NIC	Specify the IP address of the same subnet as the physical IP address of the 100GNIC that created the VF of the SR-IOV. You only need to set the same IP address as your own ipAddress (you do not need to set the subnet
inputIPAddress: 192.174.90.121	You can set the same IP address as your own ipAddress.	
inputPort: 16000	own port number	
branchOutputIPAddress: 192.174.90.141,192.174.9	Specify destination (GPU advanced inference 1, GPU advanced inference 2) IP addresses separated by commas	
branchOutputPort: 17000,18000	Specify destination (GPU advanced inference 1, GPU advanced inference 2) port numbers separated by commas	
- functionKey: infer-1	GPU advanced inference Function1 identifier	
userParams:		
ipAddress: 192.174.90.141/24	The own IP address. Set as the IP address of Pod 2nd NIC	Specify the IP address of the same subnet as the physical IP address of the 100GNIC that created the VF of the SR-IOV.
inputIPAddress: 192.174.90.141	The own IP address. Used for GStreamer video processing commands (fpgpay) (no subnet mask setting required)	You only need to set the same IP address as your own ipAddress (you do not need to set the subnet mask).
inputPort: 17000	own port number	
outputIPAddress: 192.174.90.10	Destination (Video reception tool) IP address	
outputPort: 2001	Destination (Video reception tool) port number	
- functionKey: infer-2	GPU advanced inference Function2 identifier	
userParams:		
ipAddress: 192.174.90.142/24	The own IP address. Set as the IP address of Pod 2nd NIC	Specify the IP address of the same subnet as the physical IP address of the 100GNIC that created the VF of the SR-IOV.
inputIPAddress: 192.174.90.142	The own IP address. Required for GStreamer video processing command execution (fpgpay)	You can set the same IP address as your own ipAddress.
inputPort: 18000	own port number	
outputIPAddress: 192.174.90.10	Destination (Video reception tool) IP address	
outputPort: 2002	Destination (Video reception tool) port number	
userRequirement: user-requirement	Specifies metadata.name of UserRequirement ConfigMap to be referenced to obtain various configuration information for DataFlow scheduling	For details, refer to "Section 9.9 Setting the Scheduling Strategy for DataFlow" in "OpenKasugai-Controller Install Manual"

FunctionChain YAML	Description	Remarks
apiVersion: example.com/v1		
kind: FunctionChain		
metadata:		
name: cpu-decode-cpu-filter-resize-2types-high-infer-c	Be set by the user	
namespace: chain-imgproc	Be set by the user	
spec:		
functionTypeNamespace: "wbfunc-imgproc"	FunctionType Namespace	
connectionTypeNamespace: "default"	ConnectionType Namespace	Not used in the current prototype
functions:	map of the Functions that make up FunctionChain key is the Function identifier specified in Connections From or To (a character string unique for this FunctionChain resource. The string to be used in each CR	
decode-main:	Identifier of the CPU decoding Function	
functionName: "cpu-decode"	Name, Version defined in FunctionTypeSpec	
version: "1.0.0"		
filter-resize-high-infer-main:	CPU filter/resize Function identifier	
functionName: "cpu-filter-resize-high-infer"	Name, Version defined in FunctionTypeSpec	
version: "1.0.0"		
copy-branch-main:	CPU copy branch Function identifier	
functionName: "copy-branch"	Name, Version defined in FunctionTypeSpec	
version: "1.0.0"		
infer-1:	GPU advanced inference (first) Function identifier	
functionName: "high-infer"	Name, Version defined in FunctionTypeSpec	
version: "1.0.0"		
infer-2:	GPU advanced inference (second) Function identifier	
functionName: "high-infer"	Name, Version defined in FunctionTypeSpec	
version: "1.0.0"		
connections:	List of Connections that make up FunctionChain	
- from:	Source Function Information in Connection	
functionKey: "wb-start-of-chain"	Identifier of the data source Function. Set Functions map key value	・If Connection From is a FunctionChain (FC) starting point (equivalent to a data source such as a surveillance camera), set the string to start with "wb-start-of-chain." ・When there are multiple start points such as FC with integration points inserted, a number or character string should be added after "wb-start-of-chain" to set a unique character string in this FC (Example 1: "wb-start-of-chain-1" or "wb-start-of-chain-2" Example 2: "wb-start-of-chain-xxx" or "wb-start-of-chain-yyy").
port: 0	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "decode-main"	Identifier of the data destination Function. Set Functions map key value	
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom
- from:	Source Function Information in Connection	
functionKey: "decode-main"	Identifier of the data source Function. Set Functions map key value	
port: 0	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "filter-resize-high-infer-main"	Identifier of the data destination Function. Set Functions map key value	
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom
- from:	Source Function Information in Connection	
functionKey: "filter-resize-high-infer-main"	Identifier of the data source Function. Set Functions map key value	
port: 0	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "copy-branch-main"	Identifier of the data destination Function. Set Functions map key value	
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom
- from:	Source Function Information in Connection	
functionKey: "copy-branch-main"	Identifier of the data source Function. Set Functions map key value	
port: 0	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "infer-1"	Identifier of the data destination Function. Set Functions map key value	
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom
- from:	Source Function Information in Connection	
functionKey: "copy-branch-main"	Identifier of the data source Function. Set Functions map key value	
port: 1	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "infer-2"	Identifier of the data destination Function. Set Functions map key value	
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom
- from:	Source Function Information in Connection	
functionKey: "infer-1"	Identifier of the data source Function. Set Functions map key value	
port: 0	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "wb-end-of-chain-1"	Identifier of the data destination Function. Set Functions map key value	・If the To of the Connection is the end point of the FC (equivalent to the external application receiving the processing result), set the string beginning with "wb-end-of-chain." ・When there are multiple end points such as FC with branch points inserted, a number or character string should be added after "wb-end-of-chain" to set a unique character string in this FC (Example 1: "wb-end-of-chain-1" or "wb-end-of-chain-2" Example 2: "wb-end-of-chain-xxx" or "wb-end-of-chain-yyy"). *Currently, numbers and strings after "wb-end-of-chain" are not used for control.
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom
- from:	Source Function Information in Connection	
functionKey: "infer-2"	Identifier of the data source Function. Set Functions map key value	
port: 0	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)	Not a TCP/UDP port number
to:	Destination Function information in Connection	
functionKey: "wb-end-of-chain-2"	Identifier of the data destination Function. Set Functions map key value	・If the To of the Connection is the end point of the FC (equivalent to the external application receiving the processing result), set the string beginning with "wb-end-of-chain." ・When there are multiple end points such as FC with branch points inserted, a number or character string should be added after "wb-end-of-chain" to set a unique character string in this FC (Example 1: "wb-end-of-chain-1" or "wb-end-of-chain-2" Example 2: "wb-end-of-chain-xxx" or "wb-end-of-chain-yyy"). *Currently, numbers and strings after "wb-end-of-chain" are not used for control.
port: 0	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)	Not a TCP/UDP port number
connectionTypeName: "auto"	Specify resource name or "auto " for ConnectionType	Current prototypes always specify "auto." Therefore, there is no reference to ConnectionType custom

FunctionType YAML	Description	Remarks
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-decode	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: decode	Function name of FPGA decoding in function catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-decode	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	

namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-cpu-decode	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: cpu-decode	CPU decoding function name in function catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-cpu-decode	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-filter-resize-high-infer	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: filter-resize-high-infer	Function name of FPGA filter/resize for GPU Advanced Inference in Function Catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-filter-resize-high-infer	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-filter-resize-low-infer	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: filter-resize-low-infer	FPGA filter/resize Function Names for GPU Lightweight Inference in Function Catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-filter-resize-low-infer	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-cpu-filter-resize-high-infer	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: cpu-filter-resize-high-infer	Function name of GPU advanced inference CPU filter/resize in Function Catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-cpu-filter-resize-high-infer	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-cpu-filter-resize-low-infer	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: cpu-filter-resize-low-infer	Function name of GPU lightweight inference CPU filter/resize in Function Catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-cpu-filter-resize-low-infer	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-copy-branch	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: copy-branch	copy branch function names in the function catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-copy-branch	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-glue-fdma-to-tcp	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: glue-fdma-to-tcp	Glue function names in the function catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-glue-fdma-to-tcp	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-high-infer	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: high-infer	Function names for GPU advanced inference in the function catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-high-infer	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	
---		
apiVersion: example.com/v1		
kind: FunctionType		
metadata:		
name: functype-low-infer	Be set by the user	
namespace: wbfunc-imgproc	Be set by the user	
spec:		
functionName: low-infer	Function names for GPU lightweight inference in the function catalog Value to be set for FunctionName in FunctionChain	
functionInfoCMRef:		
name: funcinfo-low-infer	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined	
namespace: wbfunc-imgproc	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined	
version: 1.0.0	The version of the Function. Used to ensure uniqueness with Name+Version	

FunctionInfo YAML	**Description**	**Remarks**
apiVersion: v1		
items:		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-decode	"funcinfo-"+ function name	
namespace: wbfunc-imgproc	Be set by the user (Assumptions specified by the administrator according to function catalog categories, etc.)	
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "alveo",	Deployable region type	
"inputInterfaceType": "dev2gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "dev2gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "fpgafunc-config-decode",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 6,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 15,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	

"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-cpu-decode		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "cpu",	Deployable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "cpufunc-config-decode",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
},		
{		
"name": "item2",	A name that refers to an element in the deployableItems array.	
"regionType": "cpu",	Deployable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "mem",	Interface type of output available when deployed to the above <regionType>	
"configName": "cpufunc-config-decode",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceTvpe> and <outputInterfaceTvpe> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 20,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-filter-resize-high-infer		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "alveo",	Deployable region type	
"inputInterfaceType": "dev25gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "mem",	Interface type of output available when deployed to the above <regionType>	
"configName": "fpgafunc-config-filter-resize-high-ir	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceTvpe> and <outputInterfaceTvpe> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
},		
{		
"name": "item2",	A name that refers to an element in the deplovableItems array.	
"regionType": "alveo",	Deplovable region type	
"inputInterfaceType": "mem",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "mem",	Interface type of output available when deployed to the above <regionType>	
"configName": "fpgafunc-config-filter-resize-high-ir	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 8,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 40,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-filter-resize-low-infer		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "alveo",	Deployable region type	
"inputInterfaceType": "dev25gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "mem",	Interface type of output available when deployed to the above <regionType>	
"configName": "fpgafunc-config-filter-resize-low-int	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceTvpe> and <outputInterfaceTvpe> above	
},		
{		
"name": "item2",	A name that refers to an element in the deployableItems array.	
"regionType": "alveo",	Deployable region type	
"inputInterfaceType": "mem",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "mem",	Interface type of output available when deployed to the above <regionType>	
"configName": "fpgafunc-config-filter-resize-low-int	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 8,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 40,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-cpu-filter-resize-high-infer		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deplovableItems array.	
"regionType": "cpu",	Deplovable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deploved to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "cpufunc-config-filter-resize-high-int	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceTvpe> and <outputInterfaceTvpe> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 15,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-cpu-filter-resize-low-infer		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "cpu",	Deployable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "cpufunc-config-filter-resize-low-infe	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		

spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 15,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-copy-branch		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "cpu",	Deployable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "cpufunc-config-copy-branch",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 15,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":10	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-glue-fdma-to-tcp		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "cpu",	Deployable region type	
"inputInterfaceType": "mem",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "cpufunc-config-glue-fdma-to-tcp",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 15,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-high-infer		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "a100",	Deployable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "gpufunc-config-high-infer",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
},		
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "a100",	Deployable region type	
"inputInterfaceType": "mem",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "gpufunc-config-high-infer",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 15,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
- apiVersion: v1		
kind: ConfigMap		
metadata:		
name: funcinfo-low-infer		
namespace: wbfunc-imgproc		
data:		
deployableItems: '[	A string value of an array whose elements are json objects consisting of the following key-values:	
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "t4",	Deployable region type	
"inputInterfaceType": "host100gether",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "gpufunc-config-low-infer",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
},		
{		
"name": "item1",	A name that refers to an element in the deployableItems array.	
"regionType": "t4",	Deployable region type	
"inputInterfaceType": "mem",	Interface type of input available when deployed to the above <regionType>	
"outputInterfaceType": "host100gether",	Interface type of output available when deployed to the above <regionType>	
"configName": "gpufunc-config-low-infer",	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
"specName": "spec1"	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above	
}		
]'		
spec: '[	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:	Not used in the current prototype
{		
"name": "spec1",	A name that refers to an element of the spec array.	
"minCore": 1,	Minimum number of resources to use	
"maxCore": 1,	Maximum number of resources to use	
"maxDataFlowsBase": 1,	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit	
"maxCapacityBase": 5,	Base Max Processing Power (fps)	
"maxInputNum": 1,	Maximum number of function inputs	
"maxOutputNum":1	Maximum number of outputs of the function	
}		
]'		
kind: List		

Describe the settings of network information (IP address and port number) used by Pod of each processing module in DataFlow YAML file of "2. YAML Description"
The following A shows the settings common to all processing modules, and the following B shows the settings for each processing module.

DataFlow YAML	Description
apiVersion: example.com/v1	
kind: DataFlow	
metadata:	
name: "df-test-3-1-1-1"	Be set by the user
namespace: "test01"	Be set by the user
spec:	
functionChainRef:	
name: "cpu-decode-cpu-filter-resize-2types-high-	FunctionChain metadata.Name used by DataFlow
namespace: "chain-imgproc"	FunctionChain metadata.Namespace used by DataFlow
requirements:	Describe the requirements that must be met during scheduling
all:	Describe requirements for function chain as a whole
capacity: 15	Describe the assumed load (fps)
functionUserParameter:	
- functionKey: decode-main	CPU decoding Function identifier
userParams:	
ipAddress: 192.174.90.101/24	The own IP address. Set as the IP address of Pod 2nd NIC
inputPort: 5004	own port number
outputIPAddress: 192.174.90.111	Destination (CPU filter/resize) IP address
outputPort: 15000	Destination (CPU filter/resize) port number
- functionKey: filter-resize-high-infer-main	CPU filter/resize Function identifier
userParams:	
ipAddress: 192.174.90.111/24	The own IP address. Set as the IP address of Pod 2nd NIC
inputPort: 15000	own port number
outputIPAddress: 192.174.90.121	Destination (copy branch) IP address
outputPort: 16000	Destination (copy branch) port number
- functionKey: copy-branch-main	copy branch Function identifier
userParams:	
ipAddress: 192.174.90.121/24	The own IP address. Set as the IP address of Pod 2nd NIC
inputIPAddress: 192.174.90.121	You can set the same IP address as your own ipAddress.
inputPort: 16000	own port number
branchOutputIPAddress: 192.174.90.141,192.174	Specify destination (GPU advanced inference 1, GPU advanced inference 2) IP addresses separated by commas
branchOutputPort: 17000,18000	Specify destination (GPU advanced inference 1, GPU advanced inference 2) port numbers separated by commas
- functionKey: infer-1	GPU advanced inference Function1 identifier
userParams:	
ipAddress: 192.174.90.141/24	The own IP address. Set as the IP address of Pod 2nd NIC
inputIPAddress: 192.174.90.141	The own IP address. Used for GStreamer video processing commands (fpgpay) (no subnet mask setting required)
inputPort: 17000	own port number
outputIPAddress: 192.174.90.10	Destination (Video reception tool) IP address
outputPort: 2001	Destination (Video reception tool) port number
- functionKey: infer-2	GPU advanced inference Function2 identifier
userParams:	
ipAddress: 192.174.90.142/24	The own IP address. Set as the IP address of Pod 2nd NIC
inputIPAddress: 192.174.90.142	The own IP address. Required for GStreamer video processing command execution (fpgpay)
inputPort: 18000	own port number
outputIPAddress: 192.174.90.10	Destination (Video reception tool) IP address
outputPort: 2002	Destination (Video reception tool) port number
userRequirement: user-requirement	Specifies metadata.name of UserRequirement ConfigMap to be referenced to obtain various configuration information for DataFlow scheduling

A. Settings common to all processing modules

・If this is a processing module running on an Pod
and the source for this is a processing module or Video stream tool running on another Pod, set the ipAddress and inputPort.
・It is itself a processing module running on Pod.
If your destination is a processing module or Video reception tool running on another Pod, set ipAddress and outputIPAddress and outputPort.

・Some processing modules set inputIPAddress for processing specific to the processing module in question.
・When setting ipAddress, inputIPAddress, and outputIPAddress,
In "8.7 Creating and Managing VFs for SR-IOV" in "OpenKasugai-Controller Install Manual"
Set the IP address of the common subnet with the physical IP address of the 100GNIC that created the VF.
・None of the above is required for FPGA Decode/FPGA filter/resize processing modules that do not run on Pod.

B. Settings for each processing module

A. For using CPU decoding

functionKey: decode-main
userParams:

(1)	ipAddress: 192.174.90.101/24	The own IP address. Set as the IP address of Pod 2nd NIC
(2)	inputPort: 5004	own port number
(3)	outputIPAddress: 192.174.90.111	Destination IP address
(4)	outputPort: 15000	Destination IP address

Addendum for (2) : The port number value is the port number of the CPU decoding processing module specified in stream tool Video Describe.
The following is an excerpt from Section 1.3.3 (1)(1-1) of the "OpenKasugai-Demo"

```
./start_gst_sender.sh /opt/video/input_4K_15fps.mp4 192.174.90.101 5004 1
${sleep_time:-3}
./start_gst_sender.sh /opt/video/input_4K_15fps.mp4 192.174.90.102 5004 1
${sleep_time:-3}
```

B. For using CPU filter/resize

functionKey: filter-resize-xxx-infer-main — Specify filter-resize-high-infer-main for advanced inference and filter-resize-low-infer-main for lightweight inference as functionKey
userParams:

(1)	ipAddress: 192.174.90.111/24	The own IP address. Set as the IP address of Pod 2nd NIC
(2)	inputPort: 15000	own port number
(3)	outputIPAddress: 192.174.90.121	Destination IP address
(4)	outputPort: 16000	Destination port number

C. For using copy branch

functionKey: copy-branch-main
userParams:

(1)	ipAddress: 192.174.90.121/24	The own IP address. Set as the IP address of Pod 2nd NIC
(2)	inputIPAddress: 192.174.90.121	The own IP address. Used to establish a TCP connection with the processing module in the previous stage (no subnet mask setting is required)
(3)	inputPort: 16000	own port number
(4)	branchOutputIPAddress: 192.174.90.141,192.174.90.14	Specify destination IP addresses separated by commas
(5)	branchOutputPort: 17000,18000	Specify destination port numbers separated by commas

Addendum for (2) : Specify the same IP address as in (1) (Describe as the subnet mask is not required).
Addendum for (4) : Comma-Separated IP Addresses for the Number of Destination Branches
Addendum for (5) : Comma-Separated Port Numbers for the Number of Destination Branches

D. When to use GPU inference

	functionKey: xxx-infer-main or infer-[n]	If DataFlow has no copy branch, specify high-infer-main for advanced inference and low-infer-main for lightweight inference as functionKey
	userParams:	If DataFlow has copy branch, specify infer1 or infer2 as functionKey, depending on how many GPU inferences are in the branch.
(1)	ipAddress: 192.174.90.141/24	The own IP address. Set as the IP address of Pod 2nd NIC
(2)	inputIPAddress: 192.174.90.141	The own IP address. Used for GStreamer video processing commands (fpgpay) (no subnet mask setting required)
(3)	inputPort: 16000	own port number
(4)	outputIPAddress: 192.174.90.10	IP address of Video reception tool
(5)	outputPort: 2001	Video reception tool port number

Addendum for (2) : Specify the same IP address as in (1) (Describe as the subnet mask is not required).
Addendum for (4) : The IP address value specifies the IP address of the 100GNIC of the K8s Master used by the Video reception tool.
Execute the following command on the K8s Master to check the IP address.

```
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether b4:96:91:9d:79:80 brd ff:ff:ff:ff:ff:ff
    inet 10.38.119.15/24 brd 10.38.119.255 scope global ens1f0
       valid_lft forever preferred_lft forever
    inet6 fe80::b696:91ff:fe9d:7980/64 scope link
       valid_lft forever preferred_lft forever


~ abbreviated ~


7: ens3f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:6d:65:35 brd ff:ff:ff:ff:ff:ff
    inet 192.174.90.10/24 brd 192.174.91.255 scope global ens3f1
       valid_lft forever preferred_lft forever
    inet6 fe80::e42:a1ff:fe6d:6535/64 scope link
       valid_lft forever preferred_lft forever
```

(5) The value of the port number indicates the port number specified in the video reception start procedure of the evaluation procedure.
For a example of sample-demo: Specify 2001 or 2002.
The following is an excerpt from (1) (1-1) of section 1.3.2 of the "OpenKasugai-Demo"

(1) Method for starting video reception
(1-1) Deployment of video reception tool and creation of reception script
(advanced inference result reception ×2).
(Only perform the first time, Step (1-1) is unnecessary from the second time onward.)

```
$ cd ~/openkasugai-controller/sample-functions/utils/rcv_video_tool/
$ kubectl create ns test
$ kubectl apply -f rcv_video_tool.yaml
$ kubectl get pod -n test    *Checking the pod name of video stream tool
$ kubectl exec -n test -it rcv-video-tool-XXX -- bash   *The xxx part is
changed according to the pod name confirmed above.
# vi start_test1.sh  *Paste the following and save it.
```

```
#!/bin/bash -x
```

```
#!/bin/bash -x
for i in `seq -w 01 02`
do
    gst-launch-1.0 -e udpsrc buffer-size=21299100 mtu=8900 port=20${i} ! 'application/x-rtp, media=(string)video, clock-rate=(int)90000, encoding-name=(string)RAW, sampling=(string)BGR, depth=(string)8, width=(string)1280, height=(string)1280, payload=(int)96' ! rtpvrawdepay ! queue ! videoconvert ! 'video/x-raw, format=(string)I420' ! openh264enc ! 'video/x-h264, stream-format=byte-stream, profile=(string)high' ! perf name=stream${i} ! h264parse ! qtmux ! filesink location=/tmp/output_st${i}.mp4 sync=false > /tmp/rcv_video_tool_st${i}.log &
done
```

```
# chmod +x start_test1.sh
# exit
```

E. When using Glue # 2 Processing modules not used by DataFlow described in YAML

functionKey: glue-fdma-to-tcp-main

userParams:

(1)	ipAddress: 192.174.90.131/24	The own IP address. Set as the IP address of Pod 2nd NIC
(2)	glueOutputIPAddress: 192.174.90.141	Destination IP address
(3)	glueOutputPort: 16000	Destination port number

F. For using FPGA Decode/FPGA filter/resize

No configuration required

B. Files that need to be edited from the supplied material

fixed region information

*For all regions used in the k8s cluster, the region type of each region must be specified.

predetermined-region.json	Description	Needs to be changed according to environment #	Remarks (such as what value to enter)
[			Set on a per-region basis
{	Information for one region		
“nodeName”: “worker0”,	Node name of the node where the region is located	○	
“deviceUUID”: “21330621T01J”,	Identification information of the device on which the region resides	○	・FPGA: Use the results of the "ls/dev/*" command. The FPGAd device will be shown as "xpcie _${FPGA-ID}". Enter the value of ${FPGA-ID}. ・For GPU: Use the "nvidia-smi -L" command. Because UUID(Example: GPU-b8b4f1f5-bf51-eaa3-6ec4-97190b7f6c98) is output for each device, Describe its value. ・For CPU: Describe "0" (Currently, each server is regarded as one virtual server, so a fixed value is acceptable.)"
“subDeviceSpecRef”: “lane0”,	Region name of the region	○	・For FPGA: Describe "lane"+ "${lanenumber}" ・For CPU/GPU: Describe the same value as deviceType
“regionType”: “alveou250-0100001c-2lanes-0nics”	Region type of the region	○	・For FPGAs: Describe in the following format: ""${device type}" + "-" + "${parent bs -id}" + "-" + "${number of lanes}" + "lanes" + "-" + "${number of NICs}" + "nics"" ・When using FPGA circuit (describe in section 0.4 of the OpenKasugai-Controller-InstallManual) provided as a sample, device type ="alveou250," parent bs -id="0100001c," number of lanes ="2," and number of NICs ="0" are fixed, so the regionType is always "alveou250-0100001c-2lanes-0nics." ・For CPU/GPU: Describe the same value as deviceType
},			
{	Information for one region		
“nodeName”: “worker0”,		○	
“deviceUUID”: “21330621T01J”,		○	
“subDeviceSpecRef”: “lane1”,		○	
“regionType”: “alveou250-0100001c-2lanes-0nics”		○	
},			
…			
{	Information for one region		
“nodeName”: “worker1”,		○	
“deviceUUID”: “21330621T00Y”,		○	
“subDeviceSpecRef”: “lane0”,		○	
“regionType”: “alveou250-0100001c-2lanes-0nics”		○	
},			
{	Information for one region		
“nodeName”: “worker1”,		○	
“deviceUUID”: “21330621T00Y”,		○	
“subDeviceSpecRef”: “lane1”,		○	
“regionType”: “alveou250-0100001c-2lanes-0nics”		○	
},			
{	Information for one region		
“nodeName”: “worker1”,		○	
“deviceUUID”: “gpu-702fb653-43a4-732d-6bc4-7b3487696c90”		○	
“subDeviceSpecRef”: “a100”,		○	
“regionType”: “a100”		○	GPU/CPU region type value is equivalent to device type (deviceType)
}			
]			

A. Files that can be used across all DF systems (no need to modify them to suit your environment)

Device Type Mapping Information

Mapping information to convert from model name to DeviceType of auto-acquired device.

***The following six types of devices (Alveo U250, NVIDIA GPU T4, NVIDIA GPU A100, Intel(R) Xeon(R) Gold 6346 CPU @ 3.10 GHz, Intel(R) Xeon(R) Gold 6348 CPU @ 2.60 GHz, Intel(R) Xeon(R) Gold 6330 CPU @ 2.00 GHz) are Describe in the sample data. Additional notes are required for other devices.**

devicetypemap.json	Description	Remarks
[{	Information for one device	
"inputDeviceType": "ALVEO U250 PQ",	Device information that can be obtained from the infrastructure	Setting the value obtained by running "Automatic Collection & CM Creation Tool Pre-Office Check Tool (DeviceInfoCheck.sh)"
"outputDeviceType": "alveou250"	DeviceType corresponding to the above device	"Alveo series" is "alveo."

},{	Information for one device	
"inputDeviceType": "Tesla T4",		Setting the value obtained by running "Automatic Collection & CM Creation Tool Pre-Office Check Tool (DeviceInfoCheck.sh)"
"outputDeviceType": "t4"		"Tesla T4" is "t4"
},{	Information for one device	
"inputDeviceType": "NVIDIA A100 80GB PCIe",		Setting the value obtained by running "Automatic Collection & CM Creation Tool Pre-Office Check Tool (DeviceInfoCheck.sh)"
"outputDeviceType": "a100"		"NVIDIA A100 80GB PCIe" is "a100"
},{	Information for one device	
"inputDeviceType": "Intel(R) Xeon(R) Gold 6346 CPU @ 3.10GHz",		Check grep model.name/proc/cpuinfo | sort -u results and configure
"outputDeviceType": "cpu"		"Intel (R) Xeon (R) Gold 6346 CPU @ 3.10GHz" = "cpu"
},{	Information for one device	
"inputDeviceType": "Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz",		Check grep model.name/proc/cpuinfo | sort -u results and configure
"outputDeviceType": "cpu"		"Intel (R) Xeon (R) Gold 6348 CPU @ 2.60GHz" = "cpu"
},{	Information for one device	
"inputDeviceType": "Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz",		Check grep model.name/proc/cpuinfo | sort -u results and configure
"outputDeviceType": "cpu"		"Intel (R) Xeon (R) Gold 6330 CPU @ 2.00GHz" = "cpu"
}]		

Region-specific information

Describe performance (deployment capacity and maximum processing capacity) for each region.

・In the case of FPGA, the performance of the region is assumed to vary depending on the device type (device model) and the child bitstream to be written, so it is necessary to describe the device type used × child bitstream. Since the CPU/GPU region is assumed to change only by the type of device at present, it is necessary to describe the type of device used.

・The following six types of regions are described in the sample data. If other regions are used, additional information is required.

・For FPGA, one type of region is described. Specifically, the Alveo U250 contains bitstreams for filter/resize.

・Two types of GPU regions are described. Specifically, regions for two types of devices (NVIDIA GPU T4, NVIDIA GPU A100) are described.

・Three CPU regions are described. Specifically, this region covers 3 devices: Intel (R) Xeon (R) Gold 6346 CPU @ 3.10 GHz, Intel (R) Xeon (R) Gold 6348 CPU @ 2.60 GHz, and Intel (R) Xeon (R) Gold 6330 CPU @ 2.00 GHz.

region-unique-info.json	Description	Remarks
[{	Region information for one device	The region where filter/resize child bitstream is written to the Alveo U250
"subDeviceSpecRef": "0100001c",	Id of the object to write to the device	For FPGA, the Id of the child bitstream to be written to the device (here, the Id of the bitstream for filter/resize).
"functionTargets":[{	Describe only the number of regions that would be created if this object was written	lane0 min
"regionName": "lane0",	Region name of the region (unique within the device)	"lane0" or "lane1" when using the sample child bitstream (filter/resize child bitstream)
"regionType": "alveou250-0100001c-2lanes-0nics",	Region type of the region	FPGA region type values are in the following format: ""${device type}" + "-" + "${parent bs -id}" + "-" + "${number of lanes}" + "lanes" + "-" + "${number of NICs}" + "nics""
"maxFunctions": 1,	Capacity to deploy Function (Circuit/Pod) in this region	Maximum number of Functions (circuits/Pod) that can be installed in the region
"maxCapacity": 40	Maximum processing capacity of the region	Units are fps. Values are provisional.
},{		lane1 min
"regionName": "lane1",		
"regionType": "alveou250-0100001c-2lanes-0nics",		
"maxFunctions": 1,		
"maxCapacity": 40		
}]		
},{	Region information for one device	Region for NVIDIA GPU T4
"subDeviceSpecRef": "Tesla T4",	Id of the object to write to the device	For GPU and CPU, describe the device type (device model) name.
"functionTargets":[{	Describe only the number of regions that would be created if this object was written	One GPU per device (one entire device)
"regionName": "t4",		"Tesla T4" is "t4"
"regionType": "t4",		The same value as the device type (deviceType) for CPU and GPU.
"maxFunctions": 110,		Maximum number of Functions (circuits/Pod) that can be installed in this region. Values are provisional.
"maxCapacity": 40		Units are fps. Values are provisional.
}]		
},{	Region information for one device	Space for NVIDIA GPU A100
"subDeviceSpecRef": "NVIDIA A100 80GB PCIe",	Id of the object to write to the device	For GPU and CPU, describe the device type (device model) name.
"functionTargets":[{	Describe only the number of regions that would be created if this object was written	One GPU per device (one entire device)
"regionName": "a100",		"NVIDIA A100 80GB PCIe" is "a100"
"regionType": "a100",		The same value as the device type (deviceType) for CPU and GPU.
"maxFunctions": 110,		Maximum number of Functions (circuits/Pod) that can be installed in this region. Values are provisional.

"maxCapacity": 120		Units are fps. Values are provisional.
}]		
},{	Region information for one device	Space for Intel (R) Xeon (R) Gold 6346 CPU @ 3.10 GHz
"subDeviceSpecRef": "Intel(R) Xeon(R) Gold 6346 CPU @ 3.10GHz",	Id of the object to write to the device	For GPU and CPU, describe the device type (device model) name.
"functionTargets":[{	Describe only the number of regions that would be created if this object was written	One CPU per device (one region for the entire device)
"regionName": "cpu",		CPU is determined by "cpu"
"regionType": "cpu",		The same value as the device type (deviceType) for CPU and GPU.
"maxFunctions": 110,		Maximum number of Functions (circuits/Pod) that can be installed in this region. Values are provisional.
"maxCapacity": 120		Units are fps. Values are provisional.
}]		
},{	Region information for one device	Space for Intel (R) Xeon (R) Gold 6348 CPU @ 2.60 GHz
"subDeviceSpecRef": "Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz",	Id of the object to write to the device	For GPU and CPU, describe the device type (device model) name.
"functionTargets":[{	Describe only the number of regions that would be created if this object was written	One CPU per device (one region for the entire device)
"regionName": "cpu",		CPU is determined by "cpu"
"regionType": "cpu",		The same value as the device type (deviceType) for CPU and GPU.
"maxFunctions": 110,		Maximum number of Functions (circuits/Pod) that can be installed in this region. Values are provisional.
"maxCapacity": 120		Units are fps. Values are provisional.
}]		
},{	Region information for one device	Space for Intel (R) Xeon (R) Gold 6330 CPU @ 2.00 GHz
"subDeviceSpecRef": "Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz",	Id of the object to write to the device	For GPU and CPU, describe the device type (device model) name.
"functionTargets":[{	Describe only the number of regions that would be created if this object was written	One CPU per device (one region for the entire device)
"regionName": "cpu",		CPU is determined by "cpu"
"regionType": "cpu",		The same value as the device type (deviceType) for CPU and GPU.
"maxFunctions": 110,		Maximum number of Functions (circuits/Pod) that can be installed in this region. Values are provisional.
"maxCapacity": 120		Units are fps. Values are provisional.
}]		
}]		

Information for identifying the FunctionType

Describe entries for the number of DeviceKind to be used

***The sample data describes four device types (alveou250, t4, a100, cpu). If other device types are used, the information for the additional device type must be added to the device type mapping information, and the information for the additional device type must be added to this information.**
Additional space information for additional devices is required. Please refer to the remarks column for how to add additional information.

functionkindmap.json	Description	Remarks
[{	Information for one type of Function type CR	Information for FPGAFunction
"deviceType": "alveou250",	key: string of DeviceType in WBFunc	Applicable DeviceType
"functionCRKind":"FPGAFunction"	value: Function CR type	If deviceType="alveou250," it is FPGA, so it is "FPGAFunction."
},{	Information for one type of Function type CR	Information for GPUFunction
"deviceType": "t4",		Applicable DeviceType (for NVIDIA GPU T4).
"functionCRKind":"GPUFunction"		If deviceType="t4," it is GPU, so it is "GPUFunction."
},{	Information for one type of Function type CR	Information for GPUFunction
"deviceType": "a100",		Applicable DeviceType (for NVIDIA GPU A 100).
"functionCRKind":"GPUFunction"		If deviceType="a100," it's a GPU, so it's "GPUFunction."
},{	Information for one type of Function type CR	Information for CPUFunction
"deviceType": "cpu",		Applicable DeviceType (for various cpu).
"functionCRKind":"CPUFunction"		If deviceType="cpu," it is CPU, so it is "CPUFunction."
}]		

Support information for specifying ConnectionType

Enter as many types of connections as you want to use.

***The sample data shows two types of connection (PCIe connection via shared memory and Ethernet connection). Additional information is required when using other connection types.**

connectionkindmap.json	Description	Remarks
[{	Information for one type of Connection CR	Information for PCIeConnection
"connectionMethod": "host-mem",	key — String of connectionMethod in WBConnection	ConnectionMethod for PCIe connection over shared memory
"connectionCRKind": "PCIeConnection"	value: Type of the Connection CR	If connectionMethod="host-mem," it is a PCIe connection, so it is "PCIeConnection."
},{	Information for one type of Connection CR	Information for EthernetConnection

"connectionMethod": "host-100gether",		ConnectionMethod for Ethernet connection
"connectionCRKind": "EthernetConnection"		If connectionMethod="host-100gether," it is an Ether connection, so it is fixed as "EthernetConnection."
}]		

Function Specific Information - Common Attributes

*Enter each function type of FPGA. *GPU functions do not need to be described at present (the direction to be described may change in the future).

***Only "filter-resize" is described in the sample data. Additional information is required when adding other FPGA functions. It is also necessary to update existing functions when performance values change.**

function-unique-info.json	Description	Remarks
[{	Information for 1Function	Information for filter/resize Function
"functionID" : 0,	Identifier of the Function (circuit/container image)	The value can be empty because it is not currently used.
"functionName" : "filter-resize",	such Function name	
"maxDataFlows": 8,	Maximum number of DF (WBFunc) that can be deployed to the function	In the case of FPGA circuit, the number of FunctionChannel IDs that can be provided simultaneously is indicated.
"maxCapacity": 40	Maximum processing power of the Function	
}]		

Function-specific - filter/resize only attribute

*The FunctionChannelIDList contains as many entries as the number of FunctionChannelIDs provided by all filter/resize functions (2 functions per FPGA device) written in the filter/resize (F/R) bitstream.

*For "Rx" and "Tx," values must be defined for each possible source or destination connection type.

filter-resize-childbs.json	Description	Remarks
{	Resource information for one lane of filter/resize bitstream	
"functionKernels":[{	Resource pool information prepared for one Function deployed on one lane	Only one Function per lane for filter/resize bitstream
"partitionName": "0",	Id of the Function	In the case of filter/resize, there is one Function per lane, so the value here is the Id of the lane (0 or 1).
"functionChannelIDList": [0,1,2,3,4,5,6,7],	List of FunctionChannelIDs provided by the function	Describe all FunctionChannel IDs provided by the Function. In the case of filter/resize, [0,1, 2, 3, 4, 5, 6, 7] can be fixed.
"functionChannelIDs":[{	Resource information for 1FunctionChannelID	Resource information provided for each FunctionChannelID assigned to each FPGAFunction.
"functionChannelID": 0,	The target FunctionChannelID	This is the value in functionChannelIDList above.
"rx":{	Receiving resource group provided for the FunctionChannelID	
"protocol":{	Describe for each protocol that can be communicated	
"TCP":{	Resource group to allocate when TCP is used as the protocol for communication with the source	
"port": 12300	Port number used to communicate with the source	
},		
"DMA":{	Resource group to allocate when the source communication protocol is DMA	
"port": 12300,	Port number used to communicate with the	
"lldmaConnectorID": 0,	Connector Id of LLDMA used for DMA communication with the source	
"dmaChannelID": 0	Channel Id of the DMA used to communicate with the source	
}		
}		
},		
"tx":{	Transmission resources provided for the above FunctionChannelID	
"protocol":{	Describe for each protocol that can be communicated	
"TCP":{	Resource group to allocate when TCP is used as the protocol for communication with the destination	
"port": 12300	Port number used to communicate with the destination	
},		
"DMA":{	Resource group to allocate when the protocol for communication with the destination is DMA	
"port": 12300,	Port number used to communicate with the destination	
"lldmaConnectorID": 0,	Connector Id of LLDMA used for DMA communication with destination	

"dmaChannelID": 0	Channel Id of the DMA to be used for DMA communication with the destination	
}		
}		
}		
},{	Resource information for 1FunctionChannelID	
"functionChannelID": 1,	The target FunctionChannelID	
"rx":{	Receiving resource group provided for the FunctionChannelID	
...		
},		
"tx":{	Transmission resource group provided for the FunctionChannelID	
...		
}		
},		
...		
}]		
},{	Resource pool information prepared for one Function deployed on one lane	Only one Function per lane for filter/resize bitstream
"partitionName": "1",		
...		
},		
]		
}		

Filter size function name identification correspondence information

*Mapping information for the FPGA to determine if filter/resize is for advanced or lightweight inference. Since it is determined by the value of the setting parameter (output frame size) to the FPGA, the setting parameter value and the corresponding inference type are described.

functionnamemap.json	Description	Remarks
{		
"sizeList":[{	Information for advanced inference filter/resize	for advanced inference
"height": 1280,	Size of the output frame height of the advanced inference filter/resize	Fixed value
"width": 1280,	Size of the output frame width of the advanced inference filter/resize	Fixed value
"functionName":"-high-infer"	String for the name of WBFunction corresponding to filter/resize for advanced inference	Fixed value
},{	Information for the lightweight inference filter/resize	for lightweight inference
"height": 416,	Size of output frame height of lightweight inference filter/resize	Fixed value
"width": 416,	Size of output frame width of lightweight inference filter/resize	Fixed value
"functionName":"-low-infer"	String to name WBFunction corresponding to Lightweight Inference filter/resize	Fixed value
}]		
}		

Mapping information between information by type and file names

*Information used internally by Automatic Collection & CM creation function to connect various types of information with their filenames.

premadefilelist.json	Description	Remarks
{		
"region-unique-info" : "region-unique-info.json",	"Region-specific information " and its filename	Fixed value
"function-unique-info" : "function-unique-info.json",	"Function-specific information-common attribute" and its file name	Fixed value
"filter-resize-childbs" : "filter-resize-childbs.json",	"Function-specific information-only attribute (for filter/resize)" and its file name	Fixed value
}		

The following configuration information files do not need to be changed according to the environment and can be used as they are.

Configuration information for GPUFunc

*Two types of configuration information are prepared: for advanced inference (gpufunc-config-high-infer.json) and for lightweight inference (gpufunc-config-low-infer.json). If you want to use an GPUFunction other than these, you need to create one.

Configuration information of the inference processing module for GPUFunction (performs advanced inference)

gpufunc-config-high-infer.json	Description	Remarks (No basic change is required for any value)
[{	Setting information for DMA communication on the input side and RTP communication on the output side	
"rxProtocol": "DMA",	Protocol on the input side	
"txProtocol":"RTP",	protocol on the output side	
"sharedMemoryMiB": 256,	Allocated size in HugePage used for PCIe connection [MegaByte]	Size to be reserved must be a power of 2
"imageURI": "localhost/localhost/gpu_infer_dma:1.0.0",	Container image name of the container to use	Container image name of the inference processing module for GPUFunc that performs advanced inference. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"envs":{	Environment variables set for the container to be used	Environment variable required for container execution of the inference processing module for GPUFunc that performs advanced inference, set in spec.containers [i] .env under the pod template
"CUDA_MPS_PIPE_DIRECTORY": "/tmp/nvidia-mps",	Full path of the directory for intercommunication between MPS functions	
"CUDA_MPS_LOG_DIRECTORY": "/tmp/nvidia-mps",	Full path to the directory for MPS logging.	
"SHMEM_SECONDARY": "1",	Information to control how the app is launched (shared memory management mode) in the DPDK used for PCIe connections.	Set "0" for the primary mode that is managed by the inference app alone, and set "1" for the secondary mode that is managed in conjunction with the PCIe controller. Set basic "1."
"HEIGHT": "1280",	Frame size (height) of the input video.	Advanced inference, so 1280.
"WIDTH": "1280"	Input frame size (width) of the input video.	Advanced inference, so 1280.
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "gfunc-hi-1",		There is only one container to start, so it can be fixed.
"workingDir": "/opt/nvidia/deepstream/deepstream-7.0",	Container working directory	The value is "/opt/nvidia/deepstream/deepstream-X.Y". " X.Y "is the version number of deepstream to use (no basic changes required)
"command": ["sh," "-c"],		The actual execution command is specified by "args."
"args":["cd /opt/DeepStream-Yolo && gst-launch-1.0 -ev fpgasrc !", "'video/x-raw,format=(string)BGR,%WIDTH%,%HEIGHT%'", "! nvvideoconvert ! 'video/x-raw(memory:NVMM), format=(string)RGBA'", "! m.sink_0 nvstreammux name=m nvbuf-memory-type=0 batch- size=1", "%WIDTH%", "%HEIGHT%", "! queue ! nvinfer config-file- path=./config_infer_primary_yoloV4_p6_th020_040.txt batch-size=1", "model-engine-file=./model_b1_gpu0_fp16.engine ! queue ! nvdsosd process-mode=1 ! nvvideoconvert !", "'video/x-raw, format=(string)BGR' ! videoconvert ! queue ! perf ! rtpvrawpay ! udpsink", "%OUTPUTIP%",	Arguments to be passed when executing the command	Executing commands and arguments for the Gstremer plug-in for lightweight inference of advanced processing modules running in containers
"securityContext":{		
"privileged": true		Value is fixed to "true"
},		
"volumeMounts":[{		
"name": "hugepage-1gi",	VolumeMount for the/dev/hugepages directory used by the ingress PCIe connection	There is only one container to start, so it can be fixed.
"mountPath": "/dev/hugepages"		Value is fixed to "/dev/hugepages"
},{		

"name": "host-nvidia-mps",	For MPS. VolumeMount for directories to communicate between MPS functions	
"mountPath": "/tmp/nvidia-mps"		Same value as environment variable "CUDA_MPS_PIPE_DIRECTORY"
},{		
"name": "dpdk",	VolumeMount for the directory used by the DPDK used by the ingress PCIe connection	
"mountPath": "/var/run/dpdk"		Value is fixed as "var-run-dpdk"
}],		
"resources":{		
"requests":{		
"memory": "32Gi"		For shared memory (huge pages). This setting corresponds to the k8s specification "When using hugepage, you must request at least one CPU or memory." As the value can be arbitrary, it is good to fix "32Gi."
},		
"limits":{		
"hugepages-1Gi": "1Gi"		Value is fixed to "1Gi"
}		
}		
}],		
"volumes":[{		
"name": "hugepage-1gi",	Volume for the/dev/hugepages directory used by the ingress PCIe connection	Same value as above volumeMounts. "hugepage-1gi"
"hostPath":		
{"path": "/dev/hugepages"}		Same value as "mountPath" in volumeMounts. "hugepage-1gi" above
},{		
"name": "host-nvidia-mps",	For MPS. Volume for directories to communicate between MPS functions	Same value as volumeMounts. "host-nvidia-mps" above
"hostPath":		
{"path": "/tmp/nvidia-mps"}		Same value as "mountPath" in volumeMounts. "host-nvidia-mps" above
},{		
"name": "dpdk",	Volume for the directory used by the DPDK used by the ingress PCIe connection	Same value as volumeMounts. "dpdk" above
"hostPath":		
{"path": "/var/run/dpdk"}		Same value as "mountPath" in volumeMounts. "dpdk" above
}],		
"hostNetwork": false,		At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		
"restartPolicy": "Always"		
}		
}		
},		
{	Setting information for TCP communication on the input side and RTP communication on the output side	
"rxProtocol": "TCP",	Protocol on the input side	
"txProtocol":"RTP",	protocol on the output side	
"imageURI": "localhost/gpu_infer_tcp:1.0.0",	Container image name of the container to use	Container image name of the inference processing module for GPUFunc that performs advanced inference. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"envs":{	Environment variables set for the container to be used	Environment variable required for container execution of the inference processing module for GPUFunc that performs advanced inference, set in spec.containers [i] .env under the pod template
"CUDA_MPS_PIPE_DIRECTORY": "/tmp/nvidia-mps",	Full path of the directory for intercommunication between MPS functions	
"CUDA_MPS_LOG_DIRECTORY": "/tmp/nvidia-mps",	Full path to the directory for MPS logging.	
"GST_PLUGIN_PATH": "/opt/nvidia/deepstream/deepstream-7.0/sample-functions/functions/gpu_infer_tcp_plugins/fpga_depayloader",	The directory containing the Gstreamer plugin in the container.	
"HEIGHT": "1280",	Frame size (height) of the input video.	Advanced inference, so 1280.
"WIDTH": "1280"	Input frame size (width) of the input video.	Advanced inference, so 1280.
},		
"template":{		
"apiVersion": "v1",		
"kind": "Pod",		

"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "gfunc-hi-1",		There is only one container to start, so it can be fixed.
"workingDir": "/opt/nvidia/deepstream/deepstream-7.0",	Container working directory	The value is "/opt/nvidia/deepstream/deepstream-X.Y". " X.Y "is the version number of deepstream to use (no basic changes required)
"command": ["sh," "-c"],		The actual execution command is specified by "args."
"args":["cd /opt/DeepStream-Yolo && gst-launch-1.0 -ev fpgadepay", "%INPUTIP%", "%INPUTPORT%", "! 'video/x-raw,format=(string)BGR,%WIDTH%,%HEIGHT%'", "! nvvideoconvert ! 'video/x-raw(memory:NVMM), format=(string)RGBA'", "! m.sink_0 nvstreammux name=m nvbuf-memory-type=0 batch-size=1", "%WIDTH%", "%HEIGHT%", "! queue ! nvinfer config-file-path=./config_infer_primary_yoloV4_p6_th020_040.txt batch-size=1", "model-engine-file=./model_b1_gpu0_fp16.engine ! queue ! nvdsosd process-mode=1 ! nvvideoconvert !", "'video/x-raw, format=(string)BGR' ! videoconvert ! queue ! perf ! rtpvrawpay ! udpsink", "%OUTPUTIP%", "%OUTPUTPORT%", "sync=true"]	Arguments to be passed when executing the command	Executing commands and arguments for the Gstremer plug-in for lightweight inference of advanced processing modules running in containers
"securityContext":{		
"privileged": true		
},		
"volumeMounts":[{		
"name": "host-nvidia-mps",	For MPS. VolumeMount for directories to communicate between MPS functions	
"mountPath": "/tmp/nvidia-mps"		Same value as environment variable "CUDA_MPS_PIPE_DIRECTORY"
}]		
}],		
"volumes":[{		
"name": "host-nvidia-mps",	For MPS. Volume for directories to communicate between MPS functions	Same as volumeMounts. "host-nvidia-mps" above
"hostPath":		
{"path": "/tmp/nvidia-mps"}		Same value as "mountPath" in volumeMounts. "host-nvidia-mps" above
}],		
"hostNetwork": false,		At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		
"restartPolicy": "Always"		
}		
}		
}]		

Configuration information of the inference processing module for GPUFunction (to implement lightweight inference)

gpufunc-config-low-infer.json	Description	Remarks (No basic change is required for any value)
[{	Setting information for DMA communication on the input side and RTP communication on the output side	
"rxProtocol": "DMA",	Protocol on the input side	
"txProtocol":"RTP",	protocol on the output side	
"sharedMemoryMiB": 256,	Allocated size in HugePage used for PCIe connection [MegaByte]	Size to be reserved must be a power of 2
"imageURI": "localhost/gpu_infer_dma:1.0.0",	Container image name of the container to use	Container image name of the inference processing module for GPUFunc that performs lightweight inference. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"envs":{	Environment variables set for the container to be used	Environment variable required for container execution of inference processing module for GPUFunc that performs lightweight inference, set in spec.containers [i] .env under pod template
"CUDA_MPS_PIPE_DIRECTORY": "/tmp/nvidia-mps",	Full path of the directory for intercommunication between MPS functions	
"CUDA_MPS_LOG_DIRECTORY": "/tmp/nvidia-mps",	Full path to the directory for MPS logging.	

"SHMEM_SECONDARY": "1",	Information to control how the app is launched (shared memory management mode) in the DPDK used for PCIe connections.	Set "0" for the primary mode that is managed by the inference app alone, and set "1" for the secondary mode that is managed in conjunction with the PCIe controller. Set basic "1."
"HEIGHT": "416",	Frame size (height) of the input video.	416 for lightweight inference
"WIDTH": "416"	Input frame size (width) of the input video.	416 for lightweight inference
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "gfunc-n02-lo-1",		There is only one container to start, so it can be fixed.
"workingDir": "/opt/nvidia/deepstream/deepstream-7.0",	Container working directory	The value is "/opt/nvidia/deepstream/deepstream-X.Y". " X.Y "is the version number of deepstream to use (no basic changes required)
"command": ["sh," "-c"],	Command to be executed in the container to be used	The actual execution command is specified by "args."
"args":["cd /opt/nvidia/deepstream/deepstream-7.0/sources/objectDetector_Yolo/ && gst-launch-1.0 -ev fpgasrc !", "'video/x-raw,format=(string)BGR,%WIDTH%,%HEIGHT%'", "! nvvideoconvert ! 'video/x-raw(memory:NVMM), format=(string)RGBA'", "! m.sink_0 nvstreammux name=m nvbuf-memory-type=0 batch-size=1", "%WIDTH%", "%HEIGHT%", "! queue ! nvinfer config-file-path=./config_infer_primary_yoloV3_tiny.txt", "batch-size=1 model-engine-file=./model_b1_gpu0_int8.engine ! queue ! nvvideoconvert !", "'video/x-raw, format=(string)BGR' ! videoconvert ! queue ! perf ! rtpvrawpay ! udpsink"	Arguments to be passed when executing the command	Executing commands and arguments for the Gstremer plug-in for lightweight inference implementation of inference processing modules running in containers
"securityContext":{		
"privileged": true		Value is fixed to "true"
},		
"volumeMounts":[{		
"name": "hugepage-1gi",	VolumeMount for the/dev/hugepages directory used by the ingress PCIe connection	Value is fixed as "hugepage-1gi"
"mountPath": "/dev/hugepages"		Value is fixed to "/dev/hugepages"
},{		
"name": "host-nvidia-mps",	For MPS. VolumeMount for directories to communicate between MPS functions	
"mountPath": "/tmp/nvidia-mps"		Same value as environment variable "CUDA_MPS_PIPE_DIRECTORY"
},{		
"name": "dpdk",	VolumeMount for the directory used by the DPDK used by the ingress PCIe connection	
"mountPath": "/var/run/dpdk"		Value is fixed as "var-run-dpdk"
}],		
"resources":{		
"requests":{		
"memory": "32Gi"		For shared memory (huge pages). This setting corresponds to the k8s specification "When using hugepage, you must request at least one CPU or memory." As the value can be arbitrary, it is good to fix "32Gi."
},		
"limits":{		
"hugepages-1Gi": "1Gi"	Page size of one hugepage	Value is fixed to "1Gi"
}		
}		
}],		
"volumes":[{		
"name": "hugepage-1gi",	Volume for the/dev/hugepages directory used by the ingress PCIe connection	Same value as above volumeMounts. "hugepage-1gi"
"hostPath":		
{"path": "/dev/hugepages"}		Same value as "mountPath" in volumeMounts. "hugepage-1gi" above
},{		

"name": "host-nvidia-mps",	For MPS. Volume for directories to communicate between MPS functions	Same value as volumeMounts. "host-nvidia-mps" above
"hostPath":		
{"path": "/tmp/nvidia-mps"}		Same value as "mountPath" in volumeMounts. "host-nvidia-mps" above
},{		
"name": "dpdk",	Volume for the directory used by the DPDK used by the ingress PCIe connection	Same value as volumeMounts. "dpdk" above
"hostPath":		
{"path": "/var/run/dpdk"}		Same value as "mountPath" in volumeMounts. "dpdk" above
}],		
"hostNetwork": false,		At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
},		
{	Setting information for TCP communication on the input side and RTP communication on the output side	
"rxProtocol": "TCP",	Protocol on the input side	
"txProtocol":"RTP",	protocol on the output side	
"imageURI": "localhost/gpu_infer_tcp:1.0.0",	Container image name of the container to use	Container image name of the inference processing module for GPUFunc that performs lightweight inference. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"envs":{	Environment variables set for the container to be used	Environment variable required for container execution of inference processing module for GPUFunc that performs lightweight inference, set in spec.containers [i] .env under pod template
"CUDA_MPS_PIPE_DIRECTORY": "/tmp/nvidia-mps",	Full path of the directory for intercommunication between MPS functions	
"CUDA_MPS_LOG_DIRECTORY": "/tmp/nvidia-mps",	Full path to the directory for MPS logging.	
"GST_PLUGIN_PATH": "/opt/nvidia/deepstream/deepstream-7.0/sample-functions/functions/gpu_infer_tcp_plugins/fpga_depayloader",	The directory containing the Gstreamer plugin in the container.	
"HEIGHT": "416",	Frame size (height) of the input video.	416 for lightweight inference
"WIDTH": "416"	Input frame size (width) of the input video.	416 for lightweight inference
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "gfunc-n02-lo-1",		There is only one container to start, so it can be fixed.
"workingDir": "/opt/nvidia/deepstream/deepstream-7.0",	Container working directory	The value is "/opt/nvidia/deepstream/deepstream-X.Y". " X.Y "is the version number of deepstream to use (no basic changes required)
"command": ["sh," "-c"],	Command to be executed in the container to be used	The actual execution command is specified by "args."
"args":["cd /opt/nvidia/deepstream/deepstream-7.0/sources/objectDetector_Yolo/ && gst-launch-1.0 -ev fpgadepay", "%INPUTIP%", "%INPUTPORT%", "! 'video/x-raw,format=(string)BGR,%WIDTH%,%HEIGHT%'", "! nvvideoconvert ! 'video/x-raw(memory:NVMM), format=(string)RGBA'", "! m.sink_0 nvstreammux name=m nvbuf-memory-type=0 batch-size=1", "%WIDTH%", "%HEIGHT%", "! queue ! nvinfer config-file-path=./config_infer_primary_yoloV3_tiny.txt", "batch-size=1 model-engine-file=./model_b1_gpu0_int8.engine ! queue ! nvvideoconvert !", "'video/x-raw, format=(string)BGR' ! videoconvert ! queue ! perf ! rtpvrawpay ! udpsink"	Arguments to be passed when executing the command	Executing commands and arguments for the Gstremer plug-in for lightweight inference implementation of inference processing modules running in containers
"securityContext":{		
"privileged": true		

}, "volumeMounts":[{		
"name": "host-nvidia-mps",	For MPS. VolumeMount for directories to communicate between MPS functions	
"mountPath": "/tmp/nvidia-mps"		Same value as environment variable "CUDA_MPS_PIPE_DIRECTORY"
}]		
}],		
"volumes":[{		
"name": "host-nvidia-mps",	For MPS. Volume for directories to communicate between MPS functions	Same as volumeMounts. "host-nvidia-mps" above
"hostPath": {"path": "/tmp/nvidia-mps"}		Same value as "mountPath" in volumeMounts. "host-nvidia-mps" above
}],		
"hostNetwork": false,		At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
}		

Configuration information for FPGAFunc

*There are two types of configuration information available, one for filter/resize (fpgafunc-config-filter-resize-high-infer.json) and the other for filter/resize (fpgafunc-config-filter-resize-low-infer.json). If you want to use an FPGAFunction other than these, you need to create one.

Separate configuration information is created for each value of the input parameter (as filter/resize is divided into two types, one for advanced inference and the other for lightweight inference). (Different input parameter values are assumed to define different FPGAFunction even in the same process.)

Configuration information for the FPGAFunc filter/resize processing module (performs processing for advanced inference)

fpgafunc-config-filter-resize-high-infer.json	Description	Remarks (No basic change is required for any value)
{		
"parentBitstream": {	Information about the parent Bitstream to use	
"file": "OpenKasugai-fpga-example-design-1.0.0-1.mcs",	File name of the parent Bitstream	
"id": "0100001c"	Bitstream ID of the parent Bitstream	
},		
"childBitstream": {	child Bitstream information to use	
"file": "OpenKasugai-fpga-example-design-1.0.0-2.bit",	Filename of child Bitstream	
"id": "0100001c"	Bitstream ID of the child Bitstream	
},		
"parameters": {	Environment variables set in child bs of the FPGA to be	Parameters set in bitstream for filter/resize processing module for advanced inference
"functions": {	Name of the module to be set	In the case of filter/resize, you only need to set parameters for the processing module in the functions module.
"i_width": 3840,	Size of the width of the input frame	3840 for both altitude and light
"i_height": 2160,	Input frame height size	2140 for both altitude and light
"o_width": 1280,	Output frame width size	1280 for advanced inference
"o_height": 1280	Output Frame Height Size	1280 for advanced inference
}		
},		
"sharedMemoryMiB": 256,	Allocated size in HugePage used for PCIe connection [MegaByte]	The reserved size must be a power of 2. (Basic "256" is fine)
"functionDedicatedInfo": "filter-resize-ch"		
}		

Configuration information of filter/resize processing module for FPGAFunc (performs processing for lightweight inference)

fpgafunc-config-filter-resize-low-infer.json	Description	Remarks (No basic change is required for any value)
{		
"parentBitstream": {	Information about the parent Bitstream to use	
"file": "OpenKasugai-fpga-example-design-1.0.0-1.mcs",	File name of the parent Bitstream	
"id": "0100001c"	Bitstream ID of the parent Bitstream	
},		
"childBitstream": {	child Bitstream information to use	
"file": "OpenKasugai-fpga-example-design-1.0.0-2.bit",	Filename of child Bitstream	
"id": "0100001c"	Bitstream ID of the child Bitstream	
},		
"parameters": {	Environment variables set in child bs of the FPGA to be	Parameters set in bitstream for filter/resize processing module for lightweight inference

"functions": {	Name of the module to be set	In the case of filter/resize, you only need to set parameters for the processing module in the functions module.
"i_width": 3840,	Size of the width of the input frame	3840 for both altitude and light
"i_height": 2140,	Input frame height size	2140 for both altitude and light
"o_width": 416,	Output frame width size	416 for lightweight inference
"o_height": 416	Output Frame Height Size	416 for lightweight inference
}		
},		
"sharedMemoryMiB": 256,	Allocated size in HugePage used for PCIe connection [MegaByte]	The reserved size must be a power of 2. (Basic "256" is fine)
"functionDedicatedInfo": "filter-resize-ch"		
}		

Configuration information for CPUFunc

*decoding (cpufunc-config-decode.json), filter/resize (cpufunc-config-filter-resize-high-infer.json) for advanced inference, filter/resize (cpufunc-config-filter-resize-low-infer.json) and copy branch (cpufunc-config-copy-branch.json) for lightweight inference, and

five types of configuration information for Glue (cpufunc-config-glue-fdma-to-tcp.json) are prepared. If you want to use an CPUFunction other than these, you need to create one.

Separate configuration information is created for each value of the input parameter. (Because different input parameter values are assumed to define different CPUFunction.)

Configuration information of the decoding module for CPUFunction

cpufunc-config-decode.json	Description	Remarks (No basic change is required for any value)
[{	Setting information for RTP communication on the input side and DMA communication on the output side	
"rxProtocol":"RTP",	Protocol on the input side	"RTP" fixed
"txProtocol":"DMA",	protocol on the output side	"DMA" fixed
"sharedMemoryMiB": 256,	Allocated size in HugePage used for PCIe connection [MegaByte]	The reserved size must be a power of 2. (Basic "256" is fine)
"imageURI": "localhost/cpu_decode:1.0.0",	Container image name of the container to use	The container image name of the decoding module for CPUFunc. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"envs":{	Environment variables set for the container to be used	Environment variable required for container execution of the decoding module for CPUFunc, set in spec.containers [i] .env under the pod template
"DECENV_APPLOG_LEVEL": "6",	Log level	The value does not need to be changed.
"DECENV_FRAME_WIDTH": "3840",	Input Video Frame Size (Width)	3840 fixed in the sample use case
"DECENV_FRAME_HEIGHT": "2160",	Frame size (height) of the input video	2160 fixed in the sample use case
"DECENV_VIDEO_CONNECT_LIMIT": "0",	Number of consecutive connections to the video source. The default value is 0, in which case it waits indefinitely.	The default value of "0" is acceptable.
"DECENV_VIDEOSRC_PROTOCOL": "RTP",	Protocol to receive	"RTP" fixed
"DECENV_OUTDST_PROTOCOL": "DMA"	Protocol to send	"DMA" fixed
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "cfunc-1",		There is only one container to start, so it can be fixed.
"command": ["sh","-c"],	Command to be executed in the container to be used	The actual execution command is specified by "args."
"args":["./sample-functions/functions/cpu_decode/build/cpu_decode-	Arguments to be passed when executing the command	Command for executing the decoding processing module for CPUFunc in a container
"securityContext":{		
"privileged": true		Value is fixed to "true"
},		
"volumeMounts":[{		
"name": "hugepage-1gi",	VolumeMount for the/dev/hugepages directory used by the ingress PCIe connection	Value is fixed as "hugepage-1gi"
"mountPath": "/dev/hugepages"		Value is fixed to "/dev/hugepages"
},{		
"name": "dpdk",	VolumeMount for the directory used by the DPDK used by the ingress PCIe connection	
"mountPath": "/var/run/dpdk"		Value is fixed as "var-run-dpdk"
}],		

"resources":{		
"requests":{		
"memory": "32Gi"		For shared memory (huge pages). This setting corresponds to the k8s specification "When using hugepage, you must request at least one CPU or memory." As the value can be arbitrary, it is good to fix "32Gi."
},		
"limits":{		
"hugepages-1Gi": "1Gi"	Page size of one hugepage	Value is fixed to "1Gi"
}		
}		
}],		
"volumes":[{		
"name": "hugepage-1gi",	Volume for the/dev/hugepages directory used by the ingress PCIe connection	Same value as above volumeMounts. "hugepage-1gi"
"hostPath":		
{"path": "/dev/hugepages"}		Same value as "mountPath" in volumeMounts. "hugepage-1gi" above
},{		
"name": "dpdk",	Volume for the directory used by the DPDK used by the ingress PCIe connection	Same value as volumeMounts. "dpdk" above
"hostPath":		
{"path": "/var/run/dpdk"}		Same value as "mountPath" in volumeMounts. "dpdk" above
}],		
"hostNetwork": false,		At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
},		
{	Setting information for RTP communication on the input side and TCP communication on the output side	
"rxProtocol":"RTP",	Protocol of the receiver (RTP connection)	"RTP" fixed
"txProtocol":"TCP",	Protocol of the sender (Eth connection)	"TCP" fixed
"imageURI": "localhost/cpu_decode:1.0.0",	Container image name of the container to use	The container image name of the decoding module for CPUFunc. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"envs":{	Environment variables set for the container to be used	Environment variable required for container execution of the decoding module for CPUFunc, set in spec.containers [i] .env under the pod template
"DECENV_APPLOG_LEVEL": "6",	Log level	The value does not need to be changed.
"DECENV_FRAME_WIDTH": "3840",	Input Video Frame Size (Width)	3840 fixed in the sample use case
"DECENV_FRAME_HEIGHT": "2160",	Frame size (height) of the input video	2160 fixed in the sample use case
"DECENV_VIDEO_CONNECT_LIMIT": "0",	Number of consecutive connections to the video source. The default value is 0, in which case it waits indefinitely.	The default value of "0" is acceptable.
"DECENV_VIDEOSRC_PROTOCOL": "RTP",	Protocol to receive	"RTP" fixed
"DECENV_OUTDST_PROTOCOL": "TCP"	Protocol to send	"TCP" fixed
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "cfunc-1",		There is only one container to start, so it can be fixed.
"command": ["sh","-c"],	Command to be executed in the container to be used	The actual execution command is specified by "args."
"args":["./sample-functions/functions/cpu_decode/build/cpu_decode-	Arguments to be passed when executing the command	The contents are CPUFunc decoding execution commands (executable file path).
"securityContext":{		
"privileged": true		Value is fixed to "true"
}		
}],		
"hostNetwork": false,	Configuring the Container Network to Use the Docker Host-Side Network Stack	At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed

} }		
}]		

Configuration information for CPUFunction for filter/resize processing module (performs processing for advanced inference)

cpufunc-config-filter-resize-high-infer.json	Description	Remarks (No basic change is required for any value)
[{	Configuration information for TCP communication with input and output	Currently, both input and output are supported only for TCP communication, so there is only configuration information for this case.
"rxProtocol":"TCP",	Protocol on the input side	"TCP" fixed
"txProtocol":"TCP",	protocol on the output side	"TCP" fixed
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"imageURI": "localhost/cpu_filter_resize:1.0.0",	Container image name of the container to use	The container image name of the decoding module for CPUFunc. The number in the tag is the version number (no basic changes required).
"envs":{	Environment variables set for the container to be used	Environment variable required for executing the container of the inference processing module for CPUFunc that performs advanced inference, and set in spec.containers [i] .env under the pod template
"FRENV_APPLOG_LEVEL": "DEBUG",	Log level	The value does not need to be changed.
"FRENV_INPUT_WIDTH": "3840",	Input Video Frame Size (Width)	3840 fixed in the sample use case
"FRENV_INPUT_HEIGHT": "2160",	Frame size (height) of the input video	2160 fixed in the sample use case
"FRENV_OUTPUT_WIDTH": "1280",	Output Video Frame Size (Width)	Advanced inference, so 1280.
"FRENV_OUTPUT_HEIGHT": "1280"	Output Video Frame Size (Height)	Advanced inference, so 1280.
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "fr",		There is only one container to start, so it can be fixed.
"command": ["python", "fr.py", "--in_port=$(FRENV_INPUT_PORT)", "--out_addr=$(FRENV_OUTPUT_IP)", "--out_port=$(FRENV_OUTPUT_PORT)", "--in_width=$(FRENV_INPUT_WIDTH)", "--in_height=$(FRENV_INPUT_HEIGHT)", "--out_width=$(FRENV_OUTPUT_WIDTH)", "--out_height=$(FRENV_OUTPUT_HEIGHT)", "--loglevel=$(FRENV_APPLOG_LEVEL)"],	Command to be executed in the container to be used	Command to execute in container to implement filter/resize processing module for CPUFunc for advanced inference
"securityContext":{		
"privileged": true		Value is fixed to "true"
}		
}],		
"hostNetwork": false,	Configuring the Container Network to Use the Docker Host-Side Network Stack	At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
}]		

Configuration information for CPUFunction for filter/resize processing module (performs processing for lightweight inference)

cpufunc-config-filter-resize-low-infer.json	Description	Remarks (No basic change is required for any value)
[{	Configuration information for TCP communication with input and output	Currently, both input and output are supported only for TCP communication, so there is only configuration information for this case.
"rxProtocol":"TCP",	Protocol on the input side	"TCP" fixed
"txProtocol":"TCP",	protocol on the output side	"TCP" fixed
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"imageURI": "localhost/cpu_filter_resize:1.0.0",	Container image name of the container to use	The container image name of the decoding module for CPUFunc. The number in the tag is the version number (no basic changes required).

"envs":{	Environment variables set for the container to be used	Environment variable required for executing the container of the inference processing module for CPUFunc that performs advanced inference, and set in spec.containers [i] .env under the pod template
"FRENV_APPLOG_LEVEL": "DEBUG",	Log level	The value does not need to be changed.
"FRENV_INPUT_WIDTH": "3840",	Input Video Frame Size (Width)	3840 fixed in the sample use case
"FRENV_INPUT_HEIGHT": "2160",	Frame size (height) of the input video	2160 fixed in the sample use case
"FRENV_OUTPUT_WIDTH": "416",	Output Video Frame Size (Width)	Advanced inference, so 1280.
"FRENV_OUTPUT_HEIGHT": "416"	Output Video Frame Size (Height)	Advanced inference, so 1280.
},		
"template":{	Pod template data to create	The environment variables (env) to be set in the container are Described outside the template (envs above).
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "fr",		There is only one container to start, so it can be fixed.
"command": ["python", "fr.py", "--in_port=$(FRENV_INPUT_PORT)", "--out_addr=$(FRENV_OUTPUT_IP)", "--out_port=$(FRENV_OUTPUT_PORT)", "--in_width=$(FRENV_INPUT_WIDTH)", "--in_height=$(FRENV_INPUT_HEIGHT)", "--out_width=$(FRENV_OUTPUT_WIDTH)", "--out_height=$(FRENV_OUTPUT_HEIGHT)", "--loglevel=$(FRENV_APPLOG_LEVEL)"],	Command to be executed in the container to be used	Command to execute in container to implement filter/resize processing module for CPUFunc for lightweight inference
"securityContext":{		
"privileged": true		Value is fixed to "true"
}		
}],		
"hostNetwork": false,	Configuring the Container Network to Use the Docker Host-Side Network Stack	At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
}]		

Configuration Information for CPUFunction for copy branch Processing Module

cpufunc-config-copy-branch.json	Description	Remarks (No basic change is required for any value)
[{	Configuration information for TCP communication with input and output	Currently, both input and output are supported only for TCP communication, so there is only configuration information for this case.
"rxProtocol":"TCP",	Protocol on the input side	"TCP" fixed
"txProtocol":"TCP",	protocol on the output side	"TCP" fixed
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"copyMemorySize": "1024",	Memory size information per region of TCP received data storage memory (Byte)	Default value "1024" Fixed
"imageURI": "localhost/cpu_copy_branch:1.0.0",	Container image name of the container to use	The container image name of copy branch processing module for CPUFunc. The number in the tag is the version number (no basic changes required).
"template":{	Pod template data to create	
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "cfunc-copy-branch-1",		There is only one container to start, so it can be fixed.
"workingDir": "/opt/openkasugai-controller/sample-functions/functions-ext/cpu_copy_branch",	Container working directory	Value is fixed to "/opt/openkasugai-controller/sample-functions/functions-ext/cpu_copy_branch"
"command": ["sh","-c"],	Command to be executed in the container to be used	The actual execution command is specified by "args."

"args":["./copy_branch", "%RECEIVING%", "%NUM%", "%FORWARDING%", "%MEMSIZE%"],	Arguments to be passed when executing the command	Copy branch processing module for CPUFunc execution command and its arguments
"securityContext":{		
"privileged": true		Value is fixed to "true"
}		
}],		
"hostNetwork": false,	Configuring the Container Network to Use the Docker Host-Side Network Stack	At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
}]		

CPUFunction configuration information for the glue (DMA to TCP conversion) processing module

cpufunc-config-glue-fdma-to-tcp.json	Description	Remarks (No basic change is required for any value)
[{	Setting information for DMA communication on the input side and TCP communication on the output side	Since it is a processing module for converting DMA to TCP, there is only setting information for this case.
"rxProtocol":"DMA",	Protocol on the input side	"DMA" fixed
"txProtocol":"TCP",	protocol on the output side	"TCP" fixed
"sharedMemoryMiB": 256,	Allocated size in HugePage used for PCIe connection [MegaByte]	The reserved size must be a power of 2. (Basic "256" is fine)
"imageURI": "localhost/cpu_glue_dma_tcp:1.0.0",	Container image name of the container to use	The container image name of the glue (DMA to TCP conversion) processing module for CPUFunc. The number in the tag is the version number (no basic changes required).
"additionalNetwork": true,	Whether the function uses the 2nd NIC	Currently, the value is fixed to "true" because 2nd NIC is assumed to be used.
"virtualNetworkDeviceDriverType": "sriov",	Driver for the virtual NW device used by the function on the 2nd NIC	Since only "sriov" is assumed at present, the setting value is fixed to "sriov."
"template":{	Pod template data to create	
"apiVersion": "v1",		
"kind": "Pod",		
"spec":{		
"containers":[{	Configuration Information for Containers Launched in Pod	
"name": "cfunc-glue-fdma-to-tcp-1",		There is only one container to start, so it can be fixed.
"workingDir": "/opt/openkasugai-controller/sample-functions/functions-ext/cpu_glue_dma_tcp",	Container working directory	Value is fixed to "/opt/openkasugai-controller/sample-functions/functions-ext/cpu_glue_dma_tcp"
"command": ["sh","-c"],	Command to be executed in the container to be used	The actual execution command is specified by "args."
"args":["./build/glue", "%FORWARDING%", "%WIDTH%", "%HEIGHT%"],	Arguments to be passed when executing the command	Command for executing the glue (DMA to TCP conversion) processing module for CPUFunc and its arguments
"securityContext":{		
"privileged": true		Value is fixed to "true"
},		
"volumeMounts":[{		
"name": "hugepage-1gi",	VolumeMount for the/dev/hugepages directory used by the ingress PCIe connection	Value is fixed as "hugepage-1gi"
"mountPath": "/dev/hugepages"		Value is fixed to "/dev/hugepages"
},{		
"name": "dpdk",	VolumeMount for the directory used by the DPDK used by the ingress PCIe connection	
"mountPath": "/var/run/dpdk"		Value is fixed as "var-run-dpdk"
}],		
"resources":{		
"requests":{		
"memory": "32Gi"		For shared memory (huge pages). This setting corresponds to the k8s specification "When using hugepage, you must request at least one CPU or memory." As the value can be arbitrary, it is good to fix "32Gi."
},		
"limits":{		
"hugepages-1Gi": "1Gi"	Page size of one hugepage	Value is fixed to "1Gi"

}		
}		
}],		
"volumes":[{		
"name": "hugepage-1gi",	Volume for the/dev/hugepages directory used by the ingress PCIe connection	Same value as above volumeMounts. "hugepage-1gi"
"hostPath":		
{"path": "/dev/hugepages"}		Same value as "mountPath" in volumeMounts. "hugepage-1gi" above
},{		
"name": "dpdk",	Volume for the directory used by the DPDK used by the ingress PCIe connection	Same value as volumeMounts. "dpdk" above
"hostPath":		
{"path": "/var/run/dpdk"}		Same value as "mountPath" in volumeMounts. "dpdk" above
}],		
"hostNetwork": false,		At present, the value can be "false" because the use of 2nd NIC is assumed.
"hostIPC": true,		Value is fixed to "true"
"restartPolicy": "Always"		Value is "Always" fixed
}		
}		
}]		

For DeviceInfo controllers (bold red text indicates changes to the environment)

crc_deviceinfo_daemonset.yaml	Description	Remarks
apiVersion: apps/v1		
kind: DaemonSet		
metadata:		
name: crc-deviceinfo-daemon		
spec:		
selector:		
matchLabels:		
app: crc-deviceinfo-daemon		
template:		
metadata:		
labels:		
app: crc-deviceinfo-daemon		
spec:		
containers:		
- name: deviceinfo-container0		
image: localhost/deviceinfo:1.0.0		
imagePullPolicy: IfNotPresent		
securityContext:		
privileged: true		
args:		
- "--kubeconfig=/kube/config"		
env:		
- name: K8S_CLUSTERNAME	Environment variable for the k8s cluster name in the environment	Fixed value
value: default		Change to match your environment's cluster name
- name: K8S_NODENAME		
valueFrom:		
fieldRef:		
fieldPath: spec.nodeName		
volumeMounts:		
- mountPath: /kube/config		
name: crc-deviceinfo-daemon		
volumes:		
- name: crc-deviceinfo-daemon		
hostPath:		
path: /etc/k8s_worker/config		
type: File		

For PCIeConnection controllers (bold red text indicates changes to the environment)

crc_pcieconnection_daemonset.yaml	Description	Remarks
apiVersion: apps/v1		
kind: DaemonSet		
metadata:		
name: crc-pcieconnection-daemon		
spec:		
selector:		
matchLabels:		
app: crc-pcieconnection-daemon		
template:		
metadata:		
labels:		
app: crc-pcieconnection-daemon		
spec:		
containers:		
- name: pcieconnection-container0		
image: localhost/pcieconnection:1.0.0		
imagePullPolicy: IfNotPresent		
securityContext:		
privileged: true		
args:		

- "--kubeconfig=/kube/config"		
env:		
- name: K8S_NODENAME		
valueFrom:		
fieldRef:		
fieldPath: spec.nodeName		
volumeMounts:		
- mountPath: /kube/config		
name: crc-pcieconnection-daemon		
- mountPath: /var/run/dpdk	Configure the directory used by the DPDK used for PCIe connections	Configure the directory used by the DPDK
name: var-run-dpdk		
- name: hugepage-1gi	Setting up a directory prepared as a hugepage	In the assumed environment, a PCIe connection to be used as a HugePage is used, so the page size for one sheet of shared memory is 1GiB. The numeric part is changed according to the page size of one sheet of shared memory set in the worker node.
mountPath: /dev/hugepages		Set the file path of the directory prepared as a hugepage
resources:		
limits:		
hugepages-1Gi: 16Gi		Size of the hugepage set by os. Basically, the value shown on the left is acceptable.
requests:		
hugepages-1Gi: 16Gi		Size of the hugepage set by os. Basically, the value shown on the left is acceptable.
memory: 1Gi		
volumes:		
- name: crc-pcieconnection-daemon		
hostPath:		
path: /etc/k8s_worker/config		
type: File		
- name: var-run-dpdk	Configure the directory used by the DPDK used for PCIe connections	
hostPath:		
path: /var/run/dpdk		Configure the directory used by the DPDK
type: DirectoryOrCreate		
- name: hugepage-1gi	Setting up a directory prepared as a hugepage	In the assumed environment, a PCIe connection to be used as a HugePage is used, so the page size for one sheet of shared memory is 1GiB. The numeric part is changed according to the page size of one sheet of shared memory set in the worker node.
hostPath:		
path: /dev/hugepages		Set the file path of the directory prepared as a hugepage
type: DirectoryOrCreate		

For EthernetConnection controllers (no special changes for the environment)

crc_ethernetconnection_daemonset.yaml	**Description**	**Remarks**
apiVersion: apps/v1		
kind: DaemonSet		
metadata:		
name: crc-ethernetconnection-daemon		
spec:		
selector:		
matchLabels:		
app: crc-ethernetconnection-daemon		
template:		
metadata:		
labels:		
app: crc-ethernetconnection-daemon		
spec:		
containers:		
- name: ethernetconnection-container0		
image: localhost/ethernetconnection:1.0.0		
imagePullPolicy: IfNotPresent		
securityContext:		
privileged: true		
args:		

- "--kubeconfig=/kube/config"		
env:		
- name: K8S_NODENAME		
valueFrom:		
fieldRef:		
fieldPath: spec.nodeName		
volumeMounts:		
- mountPath: /kube/config		
name: crc-ethernetconnection-daemon		
volumes:		
- name: crc-ethernetconnection-daemon		
hostPath:		
path: /etc/k8s_worker/config		
type: File		

For FPGAFunction controllers (no special changes for the environment)

crc_fpgafunction_daemonset.yaml	**Description**	**Remarks**
apiVersion: apps/v1		
kind: DaemonSet		
metadata:		
name: crc-fpgafunction-daemon		
spec:		
selector:		
matchLabels:		
app: crc-fpgafunction-daemon		
template:		
metadata:		
labels:		
app: crc-fpgafunction-daemon		
spec:		
containers:		
- name: fpgafunction-container0		
image: localhost/fpgafunction:1.0.0		
imagePullPolicy: IfNotPresent		
securityContext:		
privileged: true		
args:		
- "--kubeconfig=/kube/config"		
env:		
- name: K8S_NODENAME		
valueFrom:		
fieldRef:		
fieldPath: spec.nodeName		
volumeMounts:		
- mountPath: /kube/config		
name: crc-fpgafunction-daemon		
volumes:		
- name: crc-fpgafunction-daemon		
hostPath:		
path: /etc/k8s_worker/config		
type: File		

For GPUFunction controllers (no special changes for the environment)

crc_gpufunction_daemonset.yaml	**Description**	**Remarks**
apiVersion: apps/v1		
kind: DaemonSet		
metadata:		
name: crc-gpufunction-daemon		
spec:		
selector:		
matchLabels:		
app: crc-gpufunction-daemon		
template:		

metadata:		
labels:		
app: crc-gpufunction-daemon		
spec:		
containers:		
- name: gpufunction-container0		
image: localhost/gpufunction:1.0.0		
imagePullPolicy: IfNotPresent		
securityContext:		
privileged: true		
args:		
- "--kubeconfig=/kube/config"		
env:		
- name: K8S_NODENAME		
valueFrom:		
fieldRef:		
fieldPath: spec.nodeName		
volumeMounts:		
- mountPath: /kube/config		
name: crc-gpufunc-daemon		
volumes:		
- name: crc-gpufunc-daemon		
hostPath:		
path: /etc/k8s_worker/config		
type: File		

For CPUFunction controllers (no special changes for the environment)

crc_cpufunction_daemonset.yaml	**Description**	**Remarks**
apiVersion: apps/v1		
kind: DaemonSet		
metadata:		
name: crc-cpufunction-daemon		
spec:		
selector:		
matchLabels:		
app: crc-cpufunction-daemon		
template:		
metadata:		
labels:		
app: crc-cpufunction-daemon		
spec:		
containers:		
- name: cpufunction-container0		
image: localhost/cpufunction:1.0.0		
imagePullPolicy: IfNotPresent		
securityContext:		
privileged: true		
args:		
- "--kubeconfig=/kube/config"		
env:		
- name: K8S_NODENAME		
valueFrom:		
fieldRef:		
fieldPath: spec.nodeName		
volumeMounts:		
- mountPath: /kube/config		
name: crc-cpufunc-daemon		
volumes:		
- name: crc-cpufunc-daemon		
hostPath:		
path: /etc/k8s_worker/config		
type: File		