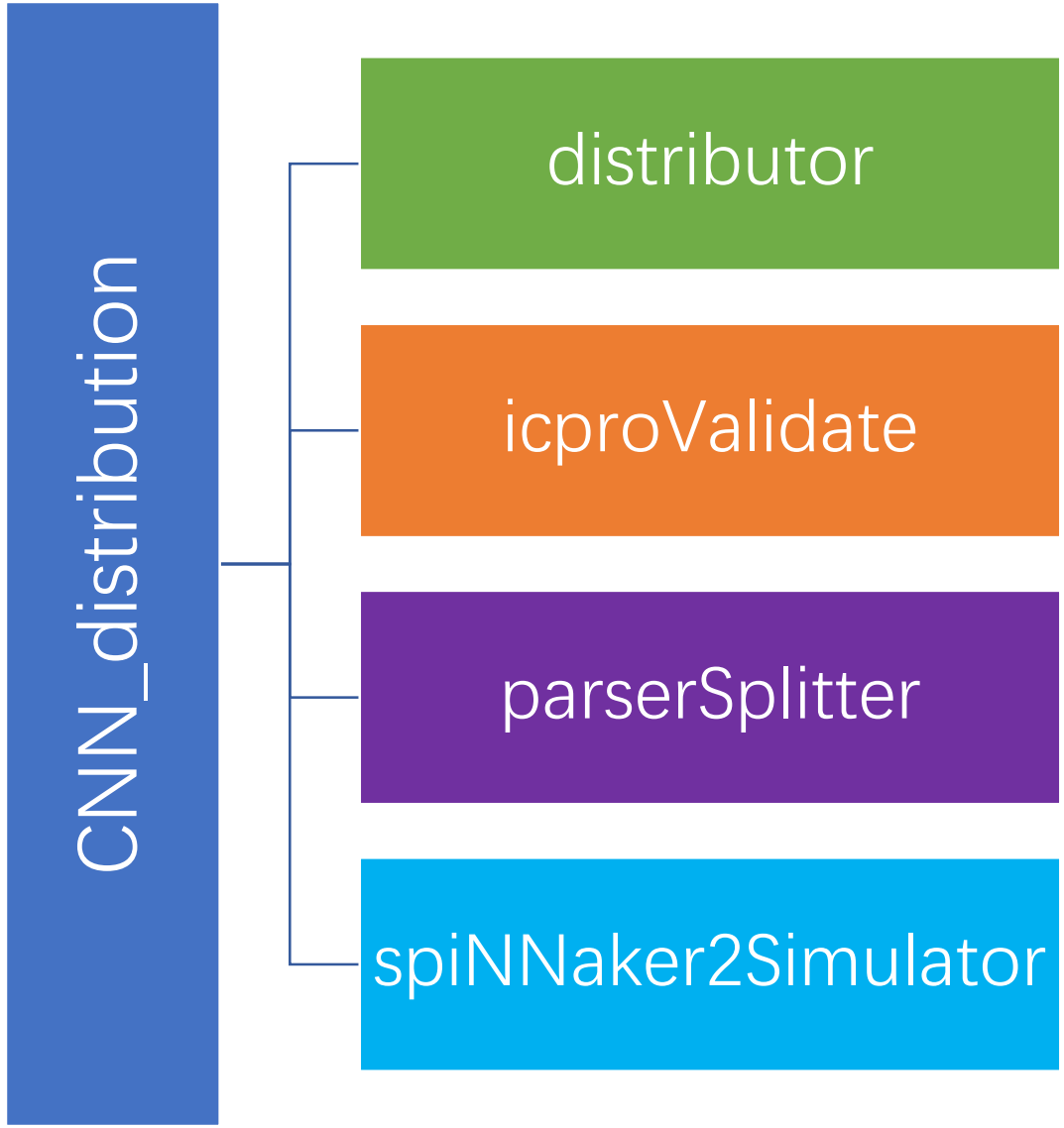
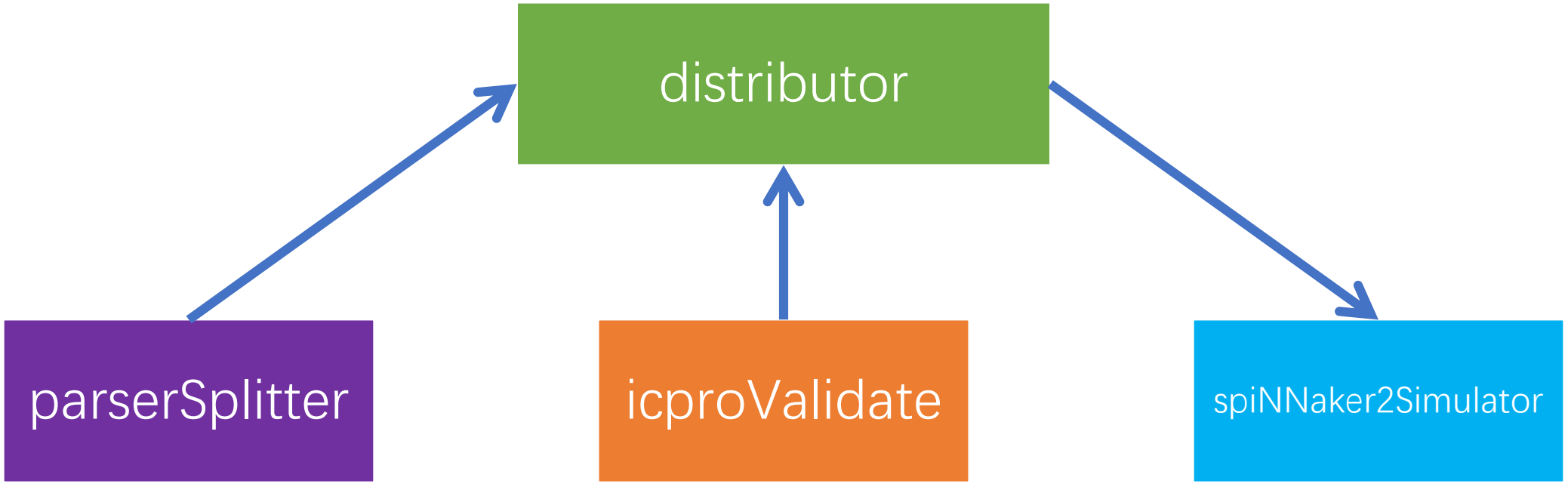


The distribution strategies for QPE with/without DRAM and SpiNNaker2 with 144 Pes are realized in this library. It depends on ***icproValidate***, ***parserSplitter*** and ***spiNNaker2Simulator***.

Library ***icproValidate*** is used to validate if the split sheme from parserSplitter is valid. Moreover, it generate validation data for ICPRO.

Library ***parserSplitter*** is for the neural network parser and splitter. The split scheme can be verified by icproValidate.

Library ***spiNNaker2Simulator*** is the simulator of SpiNNaker2.



distributor gets split scheme from ***parserSplitter***, then feeds the split scheme into ***icproValidate*** to get the data size for data transfer as well as for creating different tasks for PEs. Afterwards, ***distributor*** distribute the tasks into ***spiNNaker2Simulator***.

Type	File Name	Description
NoC	nocDataTransfer.py	Performance estimation of NoC (SpiNNaker2)
QPE-DRAM	qpeDramDistributorDataReuse.py	Distributor for CONV • With data reuse, with operator fusion
	qpeDramDistributorNoDataReuse.py	Distributor for CONV • Without data reuse, without operator fusion • Without data reuse, with Operator fusion
	qpeDramDistributorFcFusion.py	Distributor for FC • Without operator fusion • With operator fusion
	qpeDramRoofline.py	Roofline model
	qpeDramDistributionResult.txt	Distribution result
SpiNNaker2	spiNNaker2DistributorDataReuse.py	Distributor for CONV • With data reuse, with operator fusion
	spiNNaker2DistributorNoDataReuse.py	Distributor for CONV • Without data reuse, without operator fusion • Without data reuse, with Operator fusion
	spiNNaker2DistributorFcFusion.py	Distributor for FC • Without operator fusion • With operator fusion
	spiNNaker2DistributoNoneConvFc.py	Clock cycle estimation for layers, which are not CONV and FC
	spiNNaker2Roofline.py	Roofline model
	qpeDramDistributionResult.txt	Distribution result
	overallClocksBarPlot.py	Overall clocks estimation for VGG-16 and Resnet-50
Comparison of QPE-DRAM and SpiNNaker2	qpeSpiNNaker2Compare.py	Performance comparison between QPE-DRAM and SpiNNaker2
Performance report for different size of MAC arrays	SpiNNaker2HwCompareRoofline.py	Comparison of MAC arrays with different size
	SpiNNaker2_MAC_half.pdf/SpiNNaker2_MAC_half.pptx	Performance report
Other	distributionGerneral.py	Parent class for all distributor
	rooflineModel.py	Parent class for all roofline model

distributor

icproValidate

File Name	Description
dataGenerator.py	Generator data or data size for ICPRO or distributor (considering the memory alignment)
splitSchemeValidate.py	Validate if the split scheme is valid

parserSplitter

Type	File Name	Description
Parser	nnModel.py	Interface for adding different kinds of layers; **INPUT for *parserSplitter***
	nnMemParser.py	Calculate the required memory size for each layer
	nnParser.py / nnParserUpgrade.py	Parse network into multiple primitive operators
Splitter	actiLayerMapper.py	Mapping non-linearity operation into QPE/SpiNNaker2
	convLayerMapper.py, convLayerMapperDecrease.py, convLayerMapperForSpiNNaker.py	Mapping convolution operation into QPE/SpiNNaker2
	fcLayerMapper.py	Mapping matrix-multiplication operation into QPE/SpiNNaker2
	matEleLayerMapper.py	Mapping matrix element-wise operation into QPE/SpiNNaker2
	paddLayerMapper.py	Mapping padding operation into QPE/SpiNNaker2
	poolLayerMapper.py	Mapping pooling operation into QPE/SpiNNaker2
	quanLayerMapper.py	Mapping quantization operation into QPE/SpiNNaker2
Others	nnGeneral.py	Parameters and constants for parser and splitter
	nn2SpiNNaker2.py	Mapping NN to QPE/SpiNNaker2; **OUTPUT for *parserSplitter***

spiNNaker2Simulator

Type	File Name	Description
Component simulator	xxxSimulator.py	Simulator of xxx component; with data in every transferred packet. Deprecated in later development, because the simulation is very slow!
	xxxSimulatorNoDataTran.py	Simulator of xxx component but without data in every transferred packet.
triblock QPE	triBlockQpeSimulatorProcess.py	Simulator with 9 QPEs and 1 DRAM, Inheriting processing class
SpiNNaker2	spiNNaker2TriBlockSimulator.py	SpiNNaker2 simulator, creating 4 ***triBlockQpeSimulatorProcess*** process