Pacific Northwest
NATIONAL LABORATORY

MCL + Alternative Resources

PPoPP ‘22

Ryan Friese, Roberto Gioiosa, Alok Kamatar

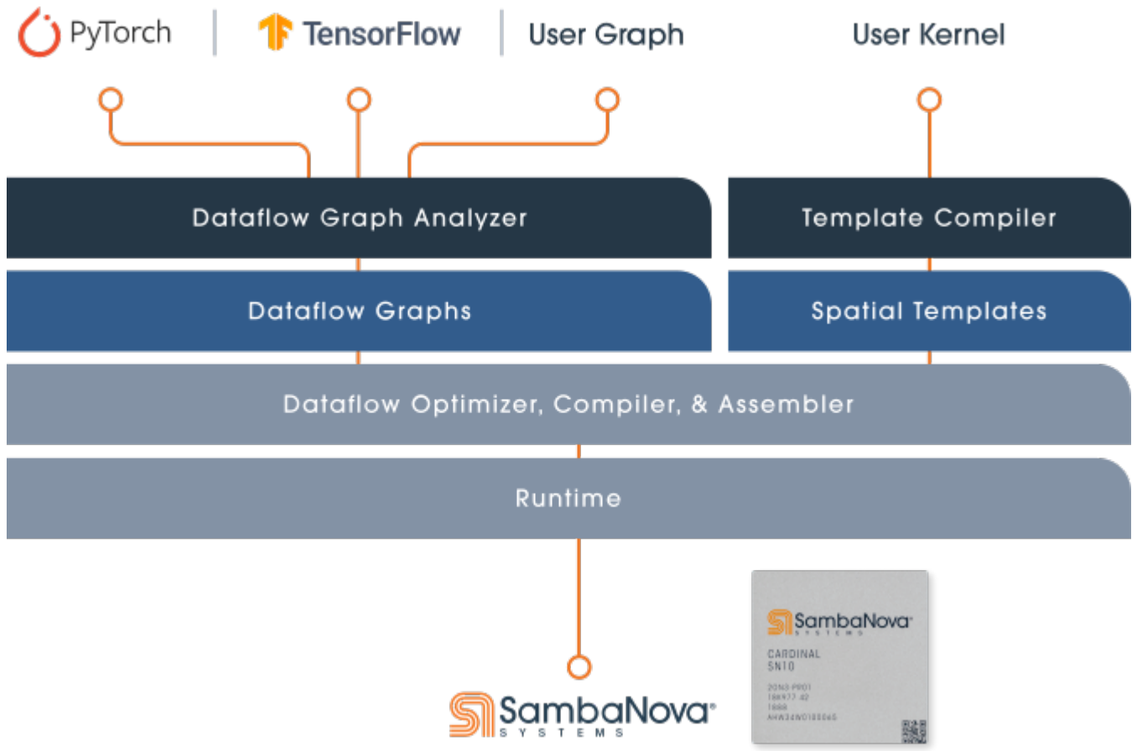
U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

Programming Alternative Resources

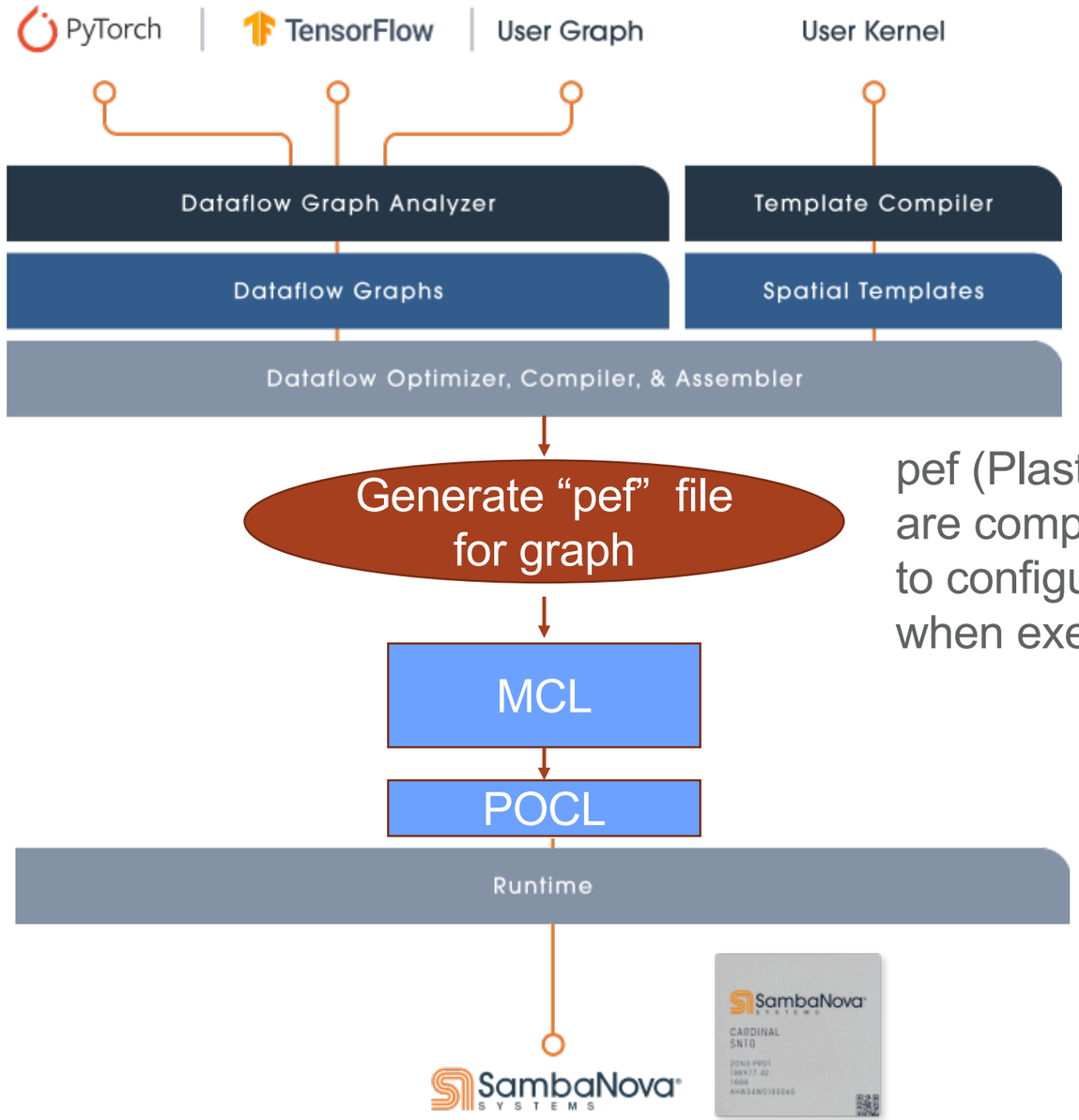
- Last year we gave a tutorial on programming an Nvida Deep Learning Accelerator (NVDLA) using MCL
- This year we present programming a SambaNova SN10
- https://sambanova.ai/

SambaNova Systems



- A dataflow architecture design to accelerate deep learning workloads
  - Software reconfigurable dataflow hardware
- Can accelerate both training and inference tasks
- Integrates with popular AI frameworks with little code modification
- Designed to be highly scalable

MCL SambaNova Stack



pef (Plasticine Executable Format) files are compiled binaries which describe how to configure the actual dataflow hardware when executing a graph

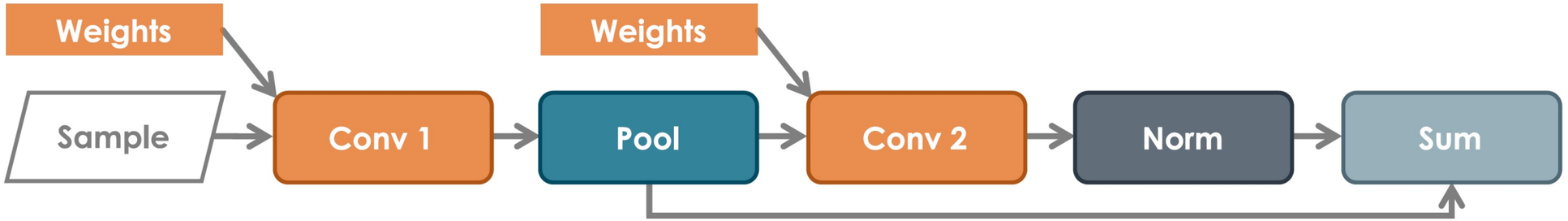
Model Construction with SambaFlow

- SambaFlow is fully integrated with popular open-source frameworks
  - E.g. TensorFlow Pytorch
- Should be able to run existing models
- Automates data and model parallel mapping to the reconfigurable hardware
- Intended to allow the programming model to scale from a single device to multiple devices and configurations

TensorFlow

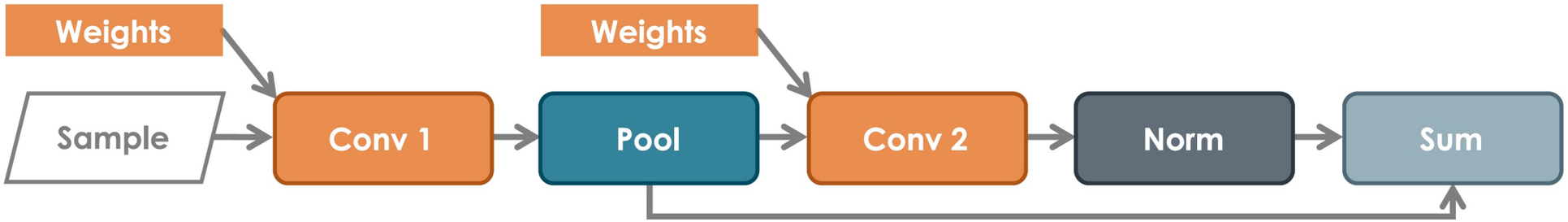
PYTORCH

User graphs



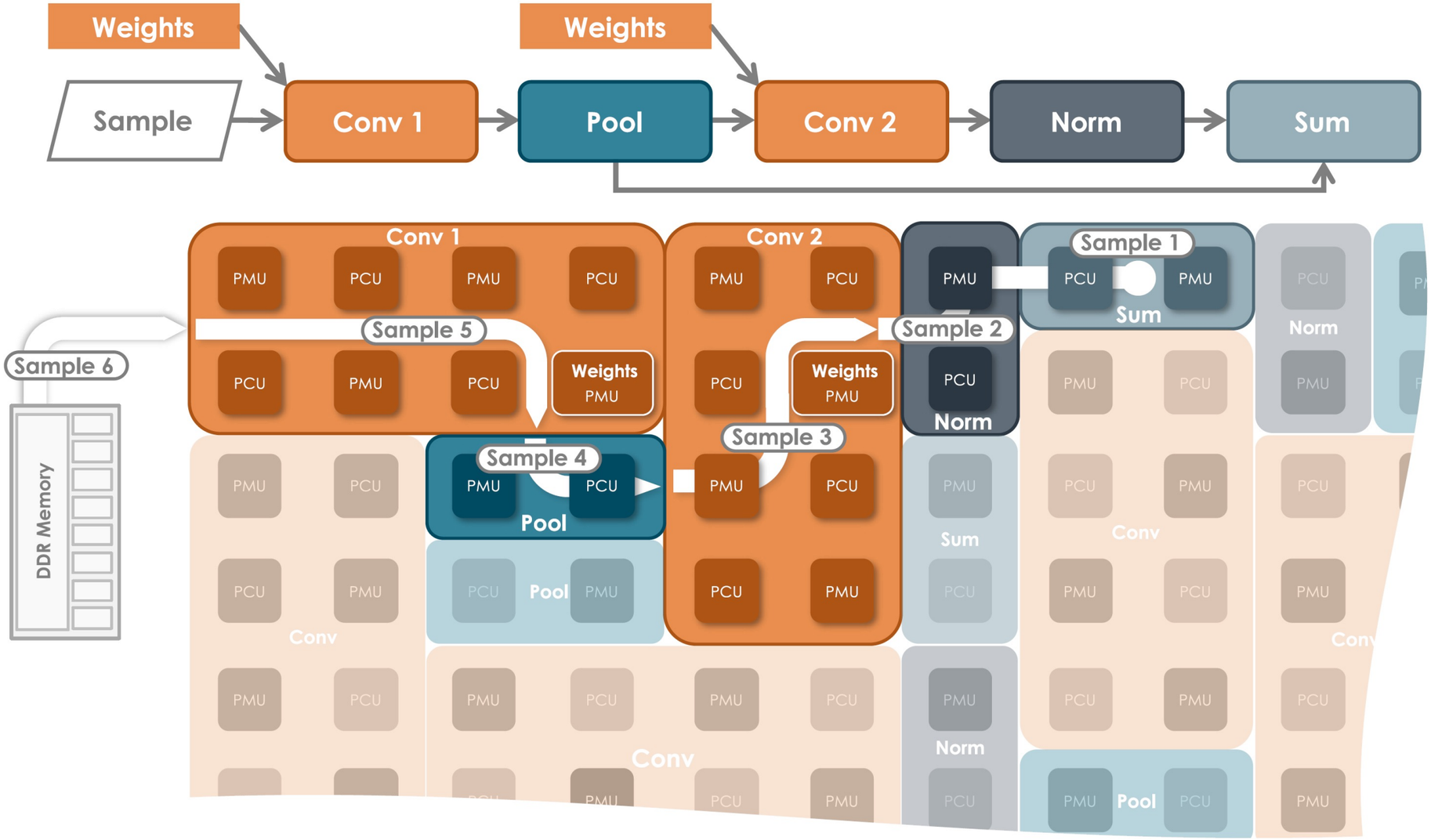
Model training and inference

- SamabaNova datascale systems are capable of accelerating both training and inference
- Both tasks require a compiled PEF binary to configure the dataflow hardware
  - Note these will be different PEF binaries



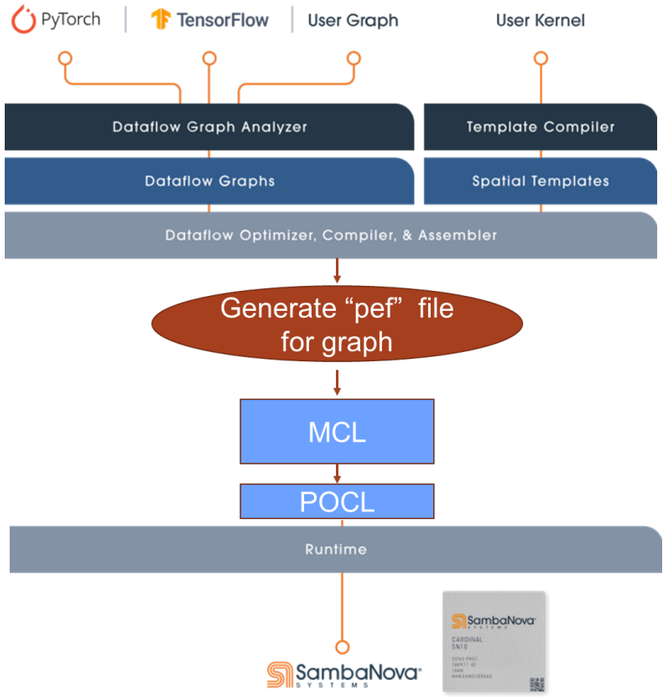
Pacific Northwest
NATIONAL LABORATORY

Mapping a model to Hardware



MCL – SambaNova integration

- Recall: MCL is built on top of OpenCL
- Recall: Tasks are OpenCL kernels (source code) and associated inputs/outputs
  - Compiled/executed depending on the device a task runs on
  - Devices managed by the MCL Scheduler
- SambaNova does not have an OpenCL implementation, nor does it compile directly from source
- For integration we have developed a custom POCL[1] device for the SambaNova
- Ingests YAML configuration files
  - PEF file name
  - Input – name, shape, dtype
  - Outuput – name, shape, dtype



[1] http://portablecl.org/

Pacific Northwest NATIONAL LABORATORY

SambaNova POCL Driver

- POCL – Portable Compute Language
  - Open source implementation of OpenCL standard
- Device discovery and initialization
- Buffer/memory management
- Launches and reports finished execution of tasks
- Implemented using the OpenCL “builtin_kernel” interface
  - Specifies that the device doesn’t run arbitrary OpenCL code
- Provides the connection between MCL and SambaNova Runtime
- Parses a user provided yaml file containing information on the PEF to load, inputs, and outputs

Sample Yaml Configuration files

- Fairly simple format
- pef: path to the pef binary
- Input: list of input buffers to the graph/model
  - Buffer name
  - Shape
  - Data type
- Output: list of output buffers to the graph/model
  - Buffer name
  - Shape
  - Data type
- Currently only one PEF per configuration file
  - In the future will be able to define multiple per file

```yaml
pef: out/mnist/minst_training.pef
input:
  - { name: image, shape: 1x784, dtype: FP32 }
  - { name: label, shape: 1x10, dtype: INT16 }
```

```yaml
pef: out/mnist/minst_inference.pef
input:
  - { name: image, shape: 1x784, dtype: FP32 }
output:
  - { name: label, shape: 1x10, dtype: INT16 }
```

Stripped Down MCL Application (MNIST)

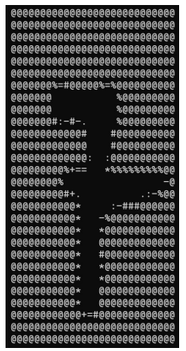
```
4   /* Handle command line args, declare variables, etc */
5
6   char* config_path= "sambanova_mnist_infer.yaml";
7   float **in; //an array of images converted to 1-D arrays of floats
8   int16_t **out; //a 10 element array indicating the class (digit) of each image
9
10  /* allocation/initialization of input/output omitted for slides */
11
12  mcl_handle** = (mcl_handle**) malloc(num_digits * rep * sizeof(mcl_handle*));
13  mcl_init(workers,0x0)
14
15  for(i=0; i < num_inferences; i++){
16      hdls[i]=mcl_task_create();
17      mcl_task_set_kernel(hdls[i],config_path,"MNIST_INFER",2,"",MCL_KERNEL_BIN);
18      mcl_task_set_arg(hdls[i], 0, (void*) in[i], sizeof(float)*IMGSIZE,
19                       MCL_ARG_INPUT | MCL_ARG_BUFFER);
20      mcl_task_set_arg(hdls[i], 1, (void*) out[i], sizeof(int16_t)*10,
21                       MCL_ARG_OUTPUT | MCL_ARG_BUFFER);
22      mcl_exec(hdls[i], pes, NULL, MCL_TASK_DF);
23  }
24
25  mcl_wait_all();
26
27  // do something with output predictions
28
29  for(i=0; i<num_inferences; i++){
30      mcl_hdl_free(hdls[i]);
31  }
```

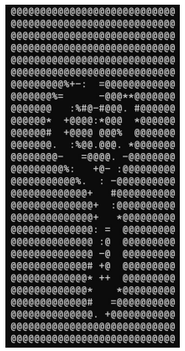
Pass yaml config path when declaring kernel

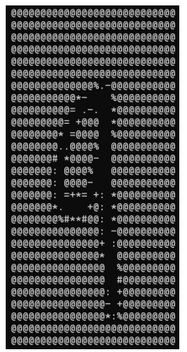
Tell MCL this is a kernel “binary”

Force execution on a dataflow Device… based on the yaml file MCL knows to use the SambaNova

- We want to perform inference to predict what digit is present within an image
- We provide MCL with a yaml file that enables us to load a PEF binary and properly initialize a context in the SambaNova runtime
  - This happens transparently for the user
- Not much difference from other MCL applications!







Pacific Northwest
NATIONAL LABORATORY

Thank you