Calorimeter Reconstruction
A Galapagos and hls4ml use case

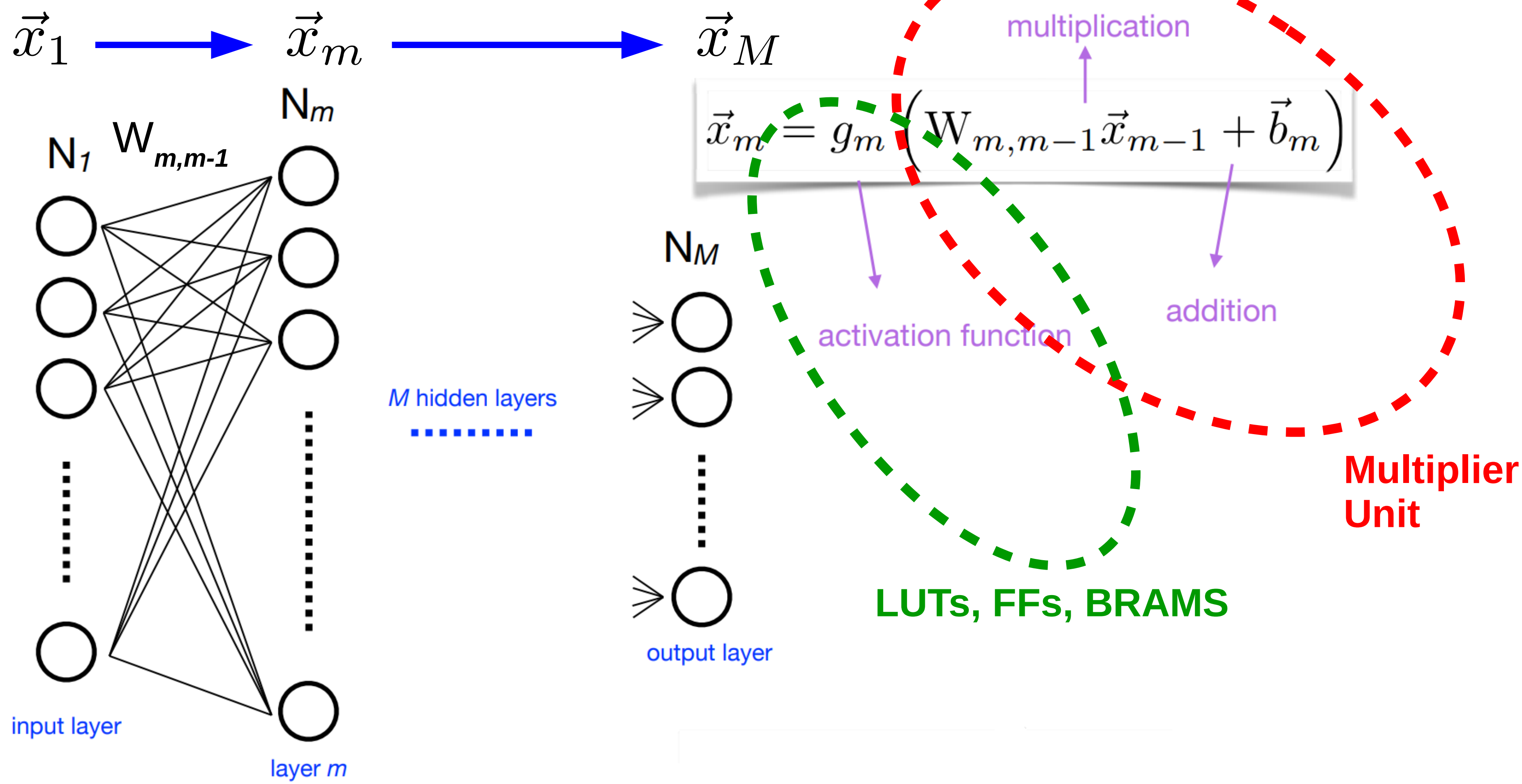
Naif Tarafdar, Paul Chow *(University of Toronto)*

Philip Harris, **Dylan Rankin**, Jeff Krupa, Sang Eon Park *(MIT)*

Fast Machine Learning Workshop

September 12th, 2019

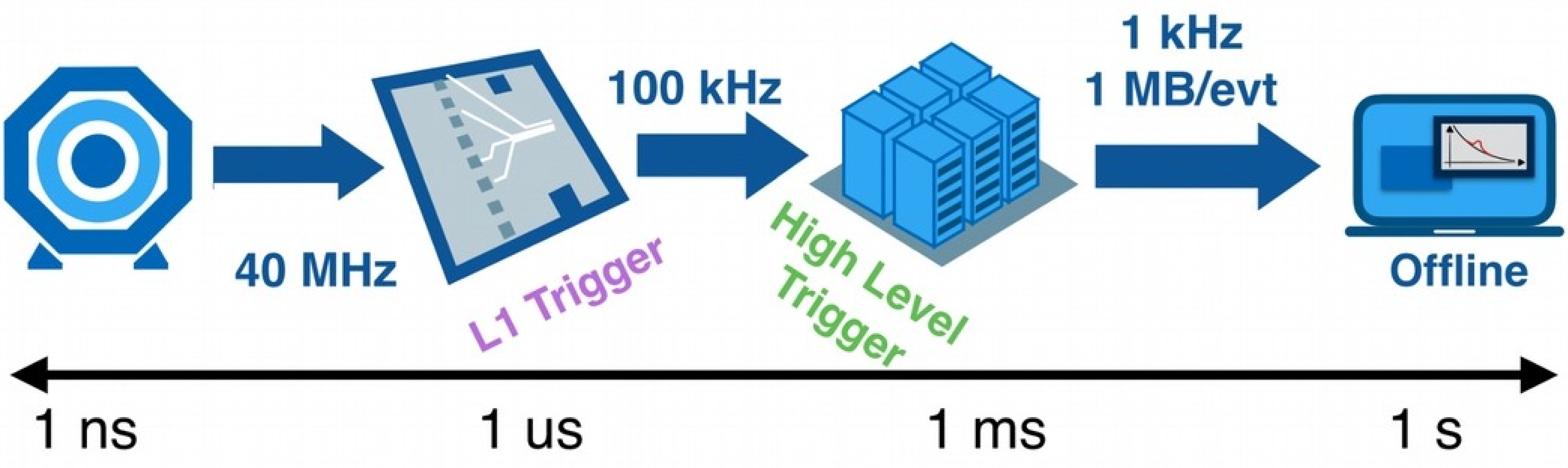
Every clock cycle
(all layer operations can be performed simultaneously)



- hls4ml tutorial:
  https://indico.cern.ch/event/822126/timetable/#3-hls4ml
- ***How can we use hls4ml, expand possibilities for usage?***

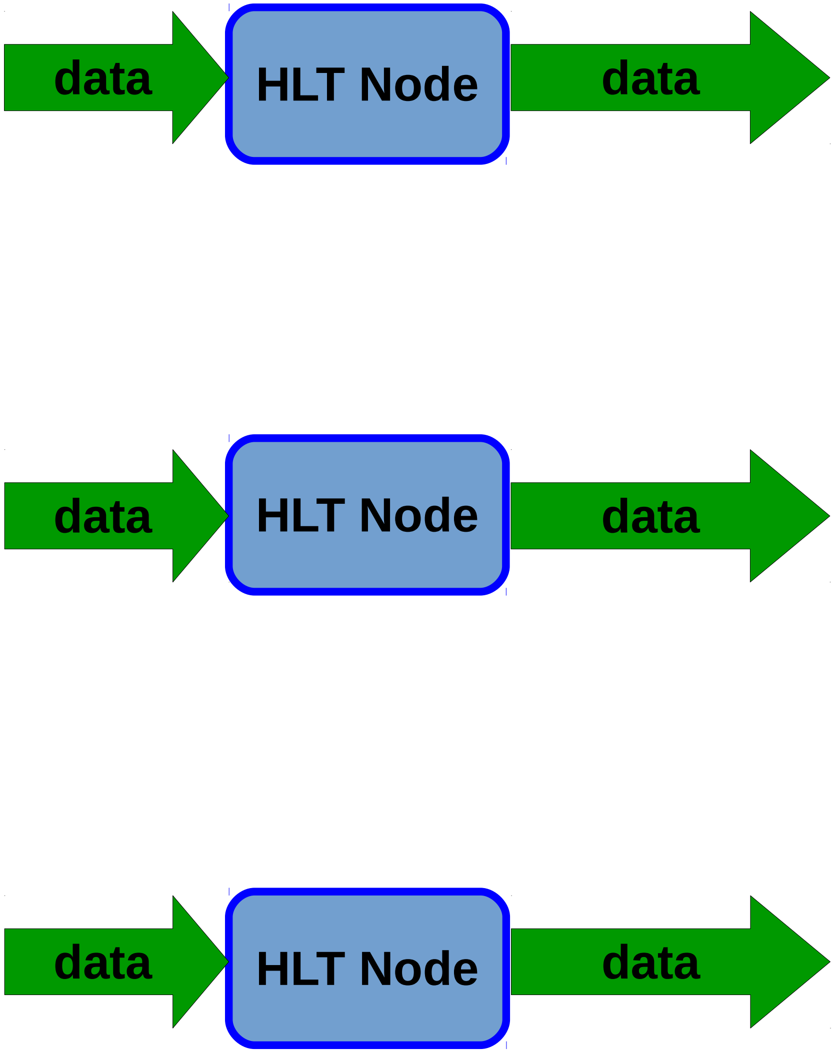
LHC Data Processing

- LHC collisions happen every 25 ns
  - 100 Tb/s in total detector data
- Must quickly select which collisions to save → FPGAs for L1
- *Can we also use FPGAs at other points in data processing?*



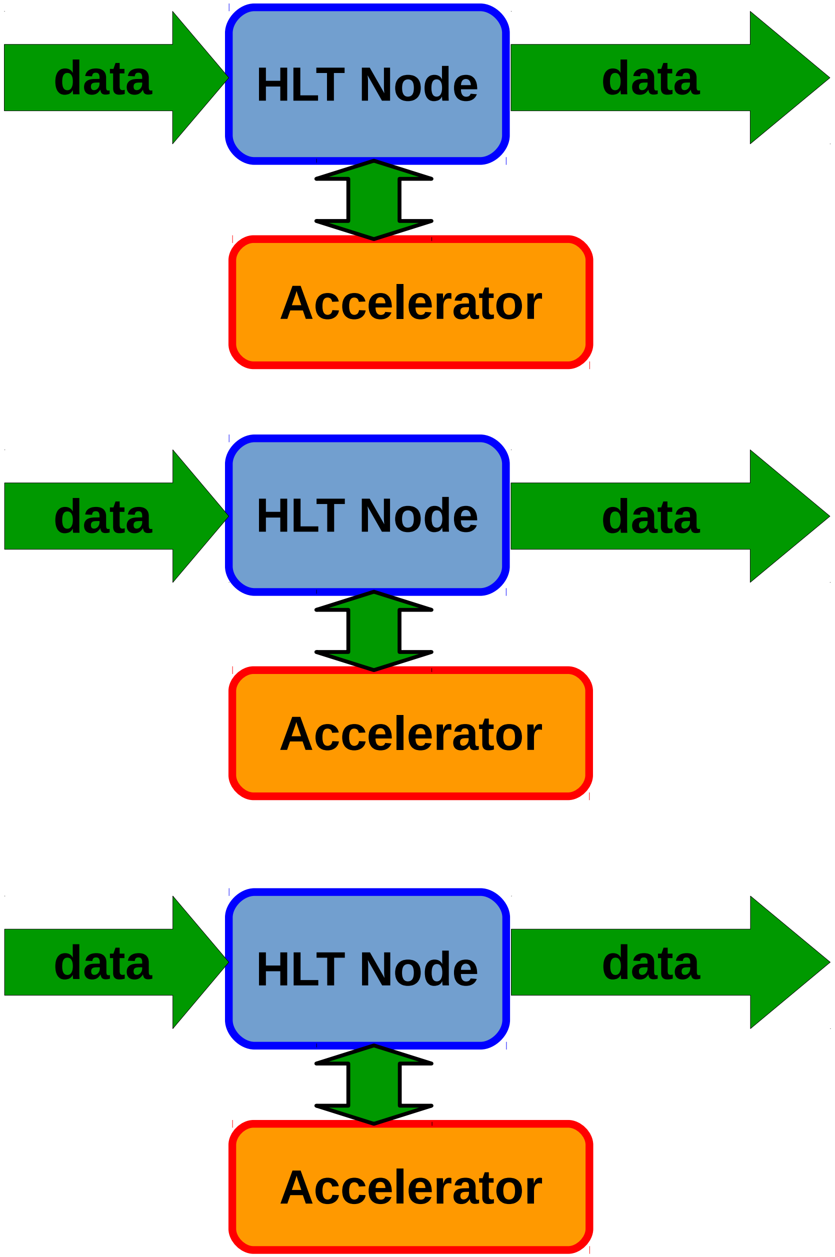
High Level Trigger

- Traditionally fully CPU-based (thousands of nodes)
- Each node processes data independently



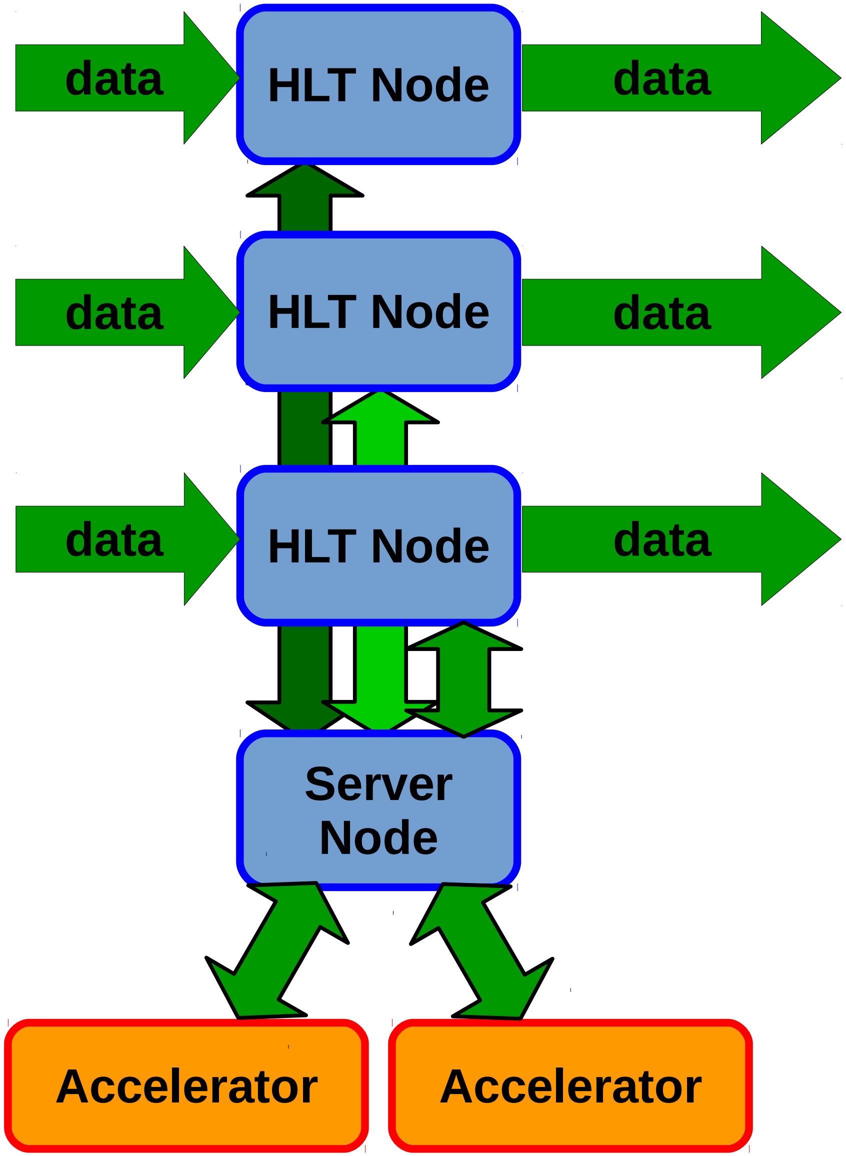
High Level Trigger

- Traditionally fully CPU-based (thousands of nodes)
- Each node processes data independently
- Recent interest in accelerators for certain large latency tasks



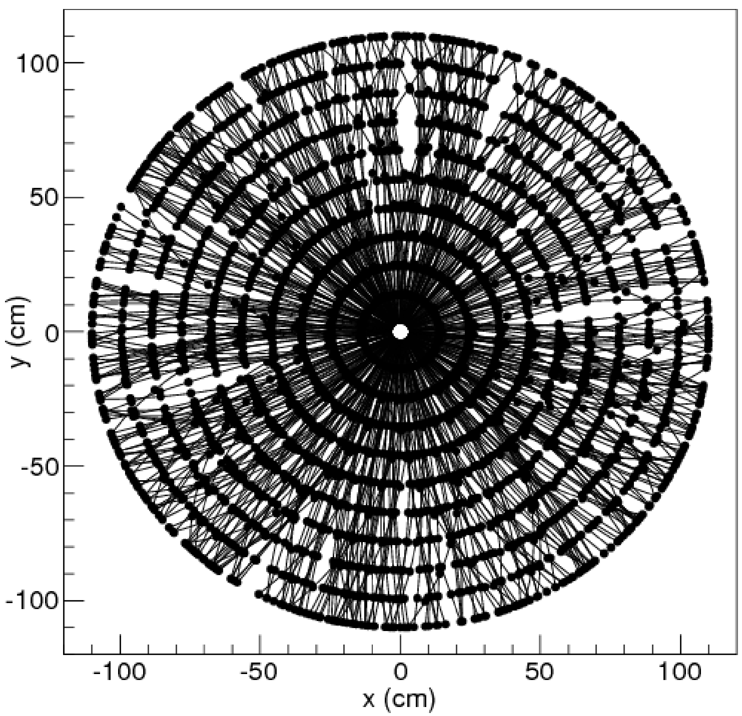
High Level Trigger

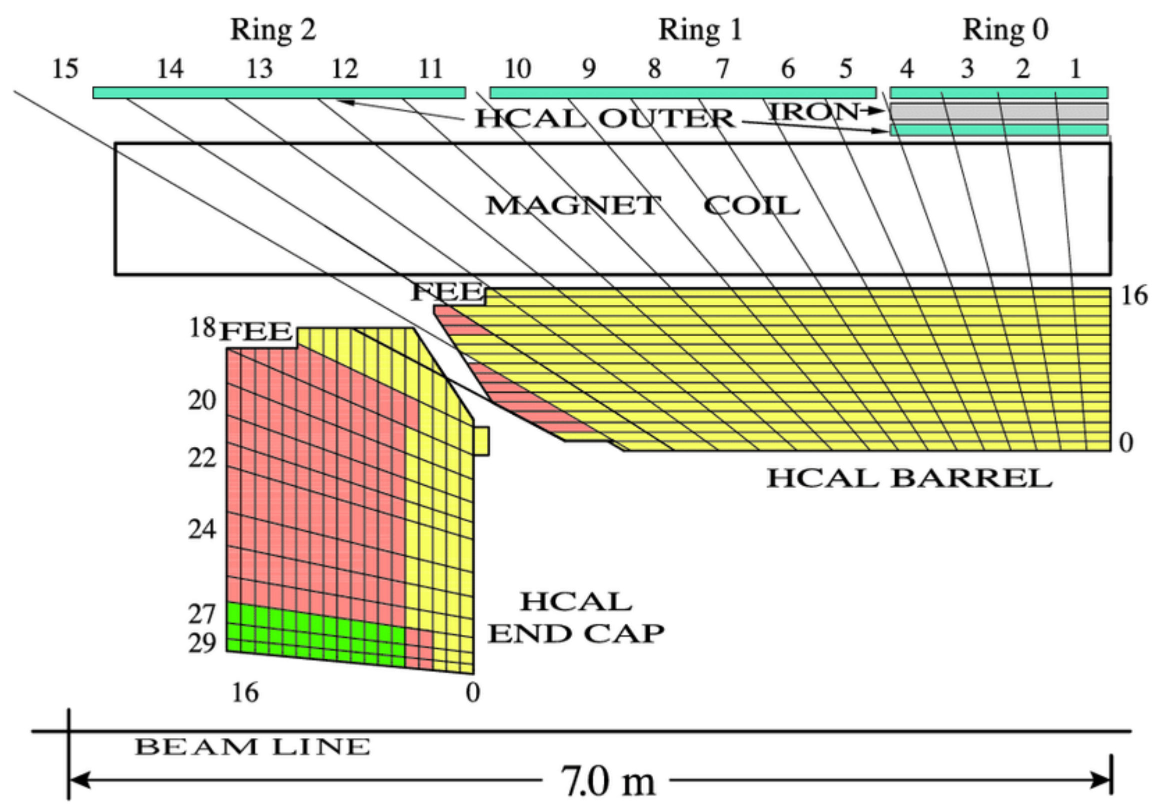
- Traditionally fully CPU-based (thousands of nodes)
- Each node processes data independently
- Recent interest in accelerators for certain large latency tasks
- Heterogeneous computing as a service offers many advantages over simpler models
  - Galapagos ideal for this design



HLT Acceleration

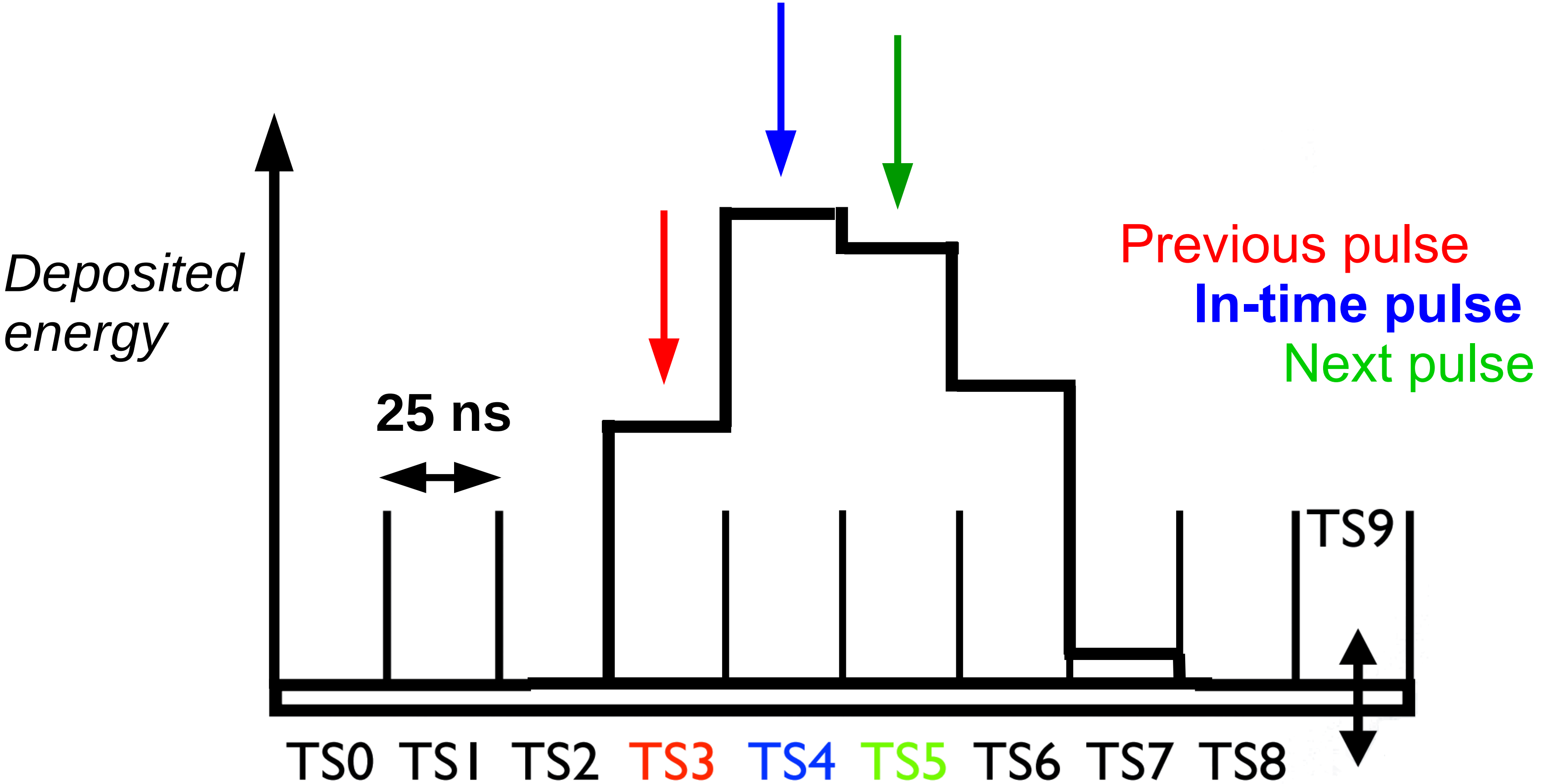
- Track reconstruction, calorimeter energy reconstruction are responsible for ~65% of all HLT processing time
  - Prime targets for acceleration





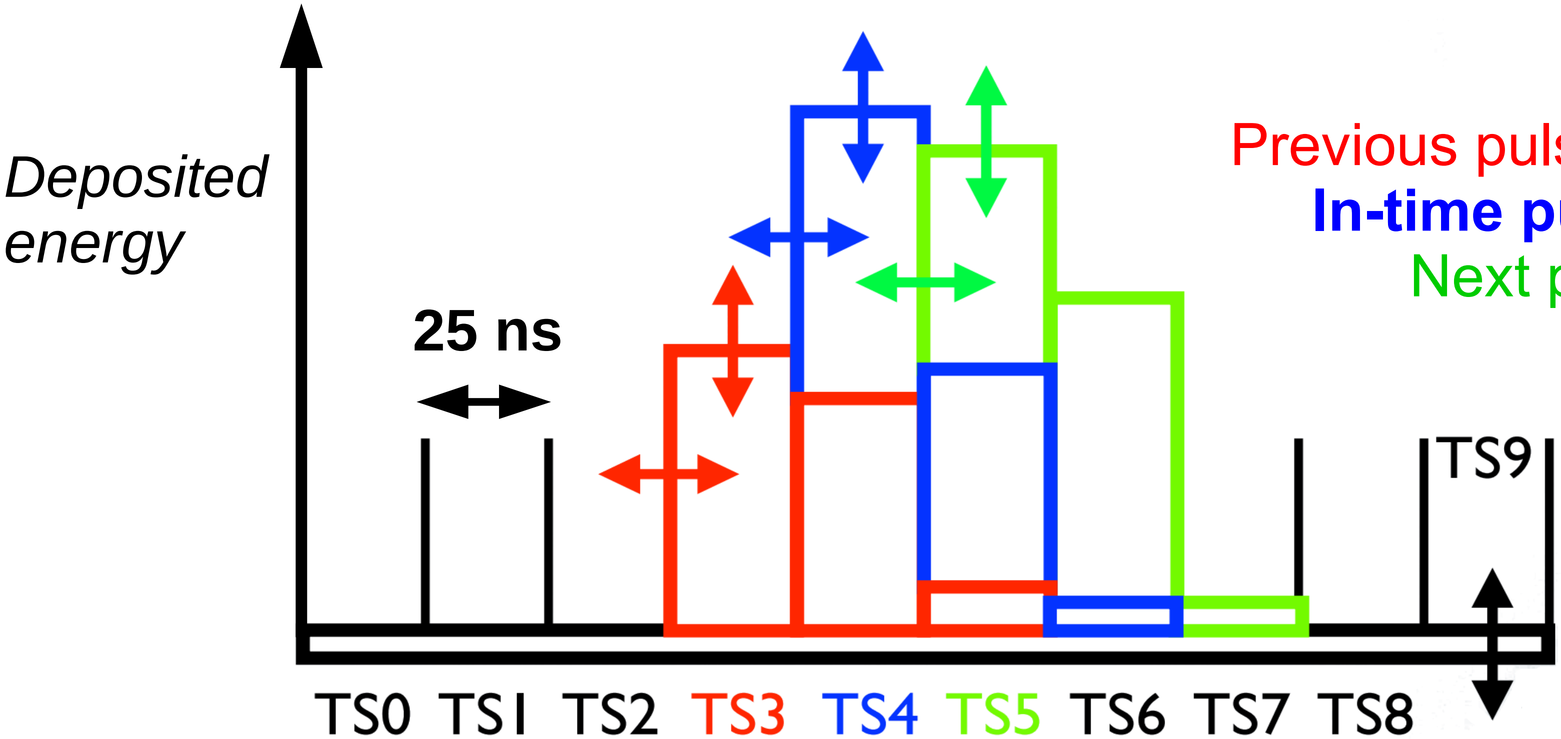
HCAL Energy Reconstruction

- Energy deposited in calorimeters from multiple collisions will overlap



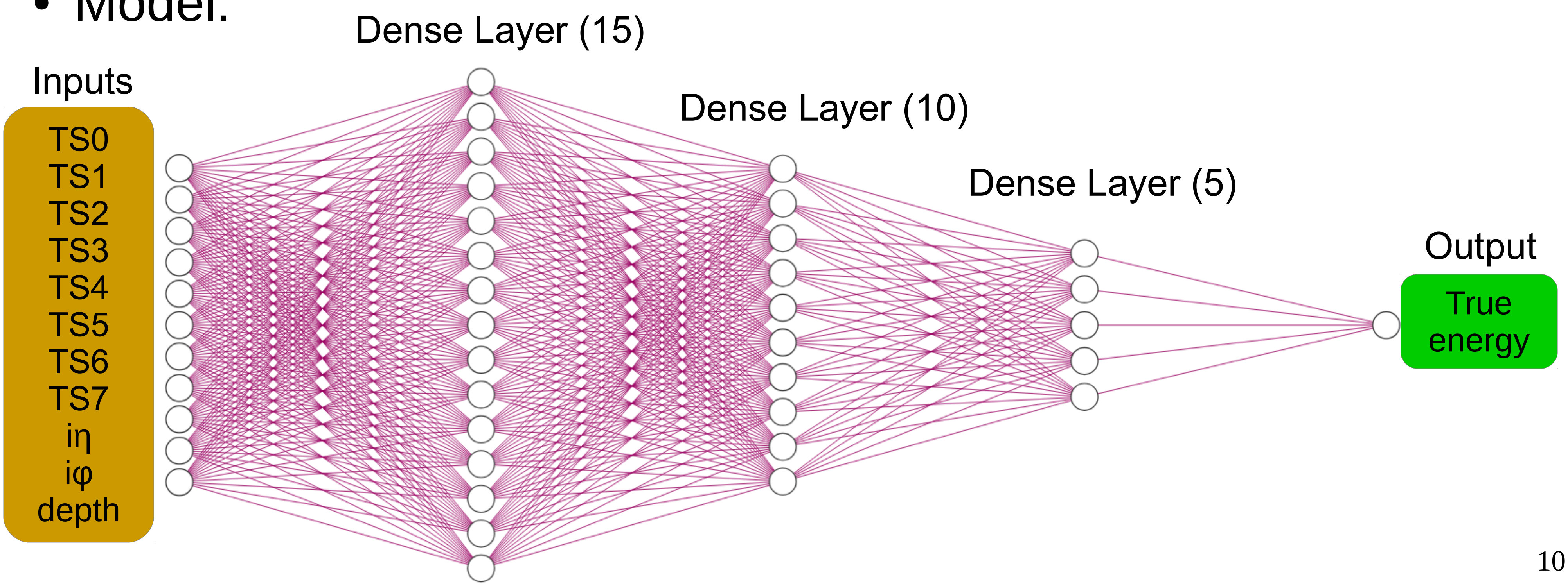
HCAL Energy Reconstruction

- Energy deposited in calorimeters from multiple collisions will overlap
- Current algorithms perform a fit of pulse shapes to extract energy of in time pulse
  - Difficult to parallelize, optimize fit for GPU/FPGA



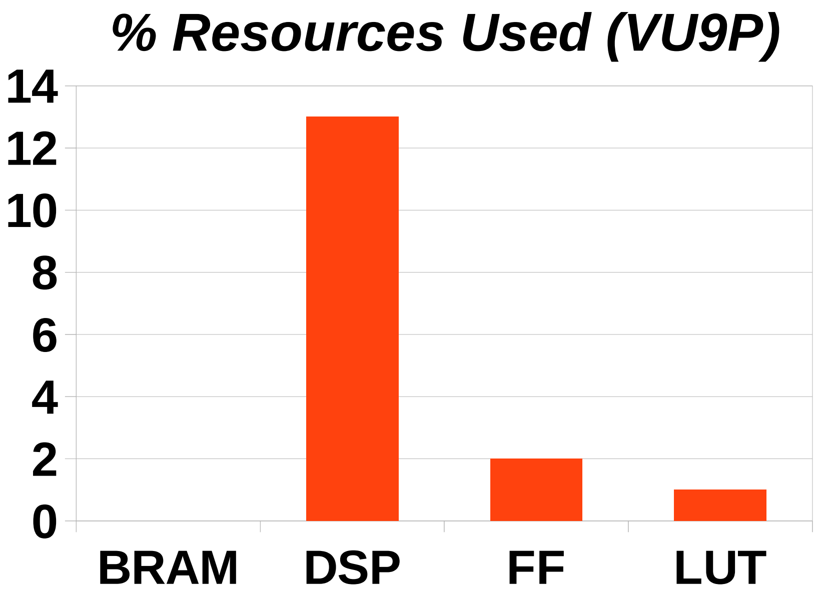
ML HCAL Reconstruction

- Machine learning provides a simple solution
  - Train a regression to the energy of the in-time pulse
- 11 Inputs : 8 raw energies (8 TS) + 3 location identifiers, 3 hidden layers (15, 10, 5 nodes)
  - Network is quite small (391 parameters)
- Model:



Inference on an FPGA

- Network implemented using hls4ml → 70 ns latency, new inference can start every 5 ns
  - 16k inferences: 80 us latency
  - Resource usage minimal, network would fit on Virtex 7



- Running on AWS (VU9P):
  - Including data transfers between FPGA ↔ CPU, total latency for 16k inferences is ~2 ms
  - Major speedup with respect to current algorithm
  - Requires usage of SDAccel, some limitations
- Running with galapagos would allow customization for specific needs, full control of FPGA ↔ CPU

Galapagos + hls4ml

- Additional files required to run hls4ml with galapagos:
  - Wrapper to handle streamed inputs in proper format
  - Definitions for galapagos
  - System configuration
  - Build scripts

```
(hls4ml-env) drankin@agent-2:~/testing/galapagos/hls4ml$ ls my-hls-test
Makefile                 generate_gal_send.tcl    myproject_test.cpp
build_prj.tcl            heterogeneous_node.cpp   program_fpga.tcl
cpu_node.cpp             heterogeneous_node.exe   tb_data
firmware                 middlewareInput          vivado_hls.log
galapagos_kerns.o        myproject.o
generate_gal_nn.tcl      myproject_prj
(hls4ml-env) drankin@agent-2:~/testing/galapagos/hls4ml$ ls my-hls-test/fi
rmware/
defines.h                inputs.h       nnet_utils    weights
galapagos_kerns.cpp      myproject.cpp  packet.h
galapagos_kerns.h        myproject.h    parameters.h
```

BACKUP