Fast inference using FPGAs for DUNE data reconstruction

24th International Conference on Computing in High-Energy and Nuclear Physics

Manuel Rodriguez

for the DUNE Collaboration

7 Nov 2019

WHAT IS DUNE?

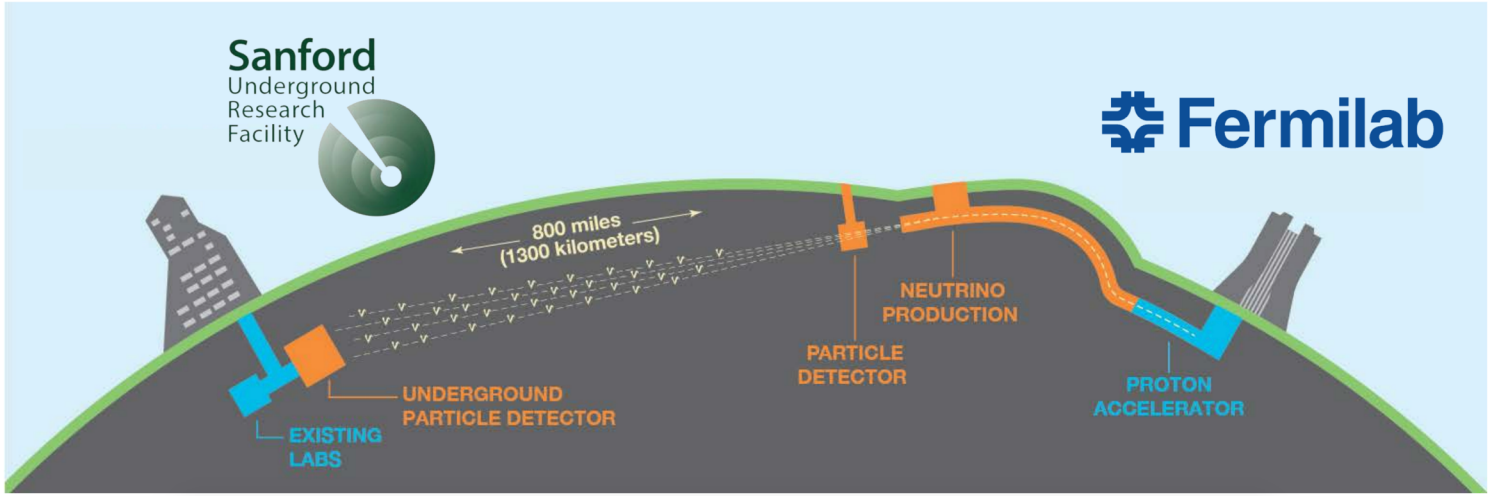
Deep Underground Neutrino Experiment

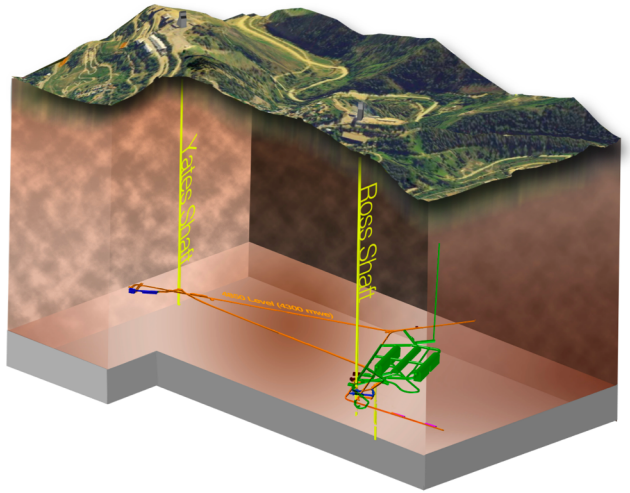
DEEP UNDERGROUND NEUTRINO EXPERIMENT

- Leading-edge, international experiment for long-baseline neutrino oscillation studies, and for neutrino astrophysics and proton decay searches.

- DUNE aims at answering fundamental questions related to:
  - the matter-antimatter asymmetry (neutrino oscillations and mass ordering).
  - the Grand Unification of forces (proton decay searches).
  - the supernova explosion mechanism (supernova neutrino detection).

More details in Heidi's plenary talk: https://indico.cern.ch/event/773049/contributions/3581360/



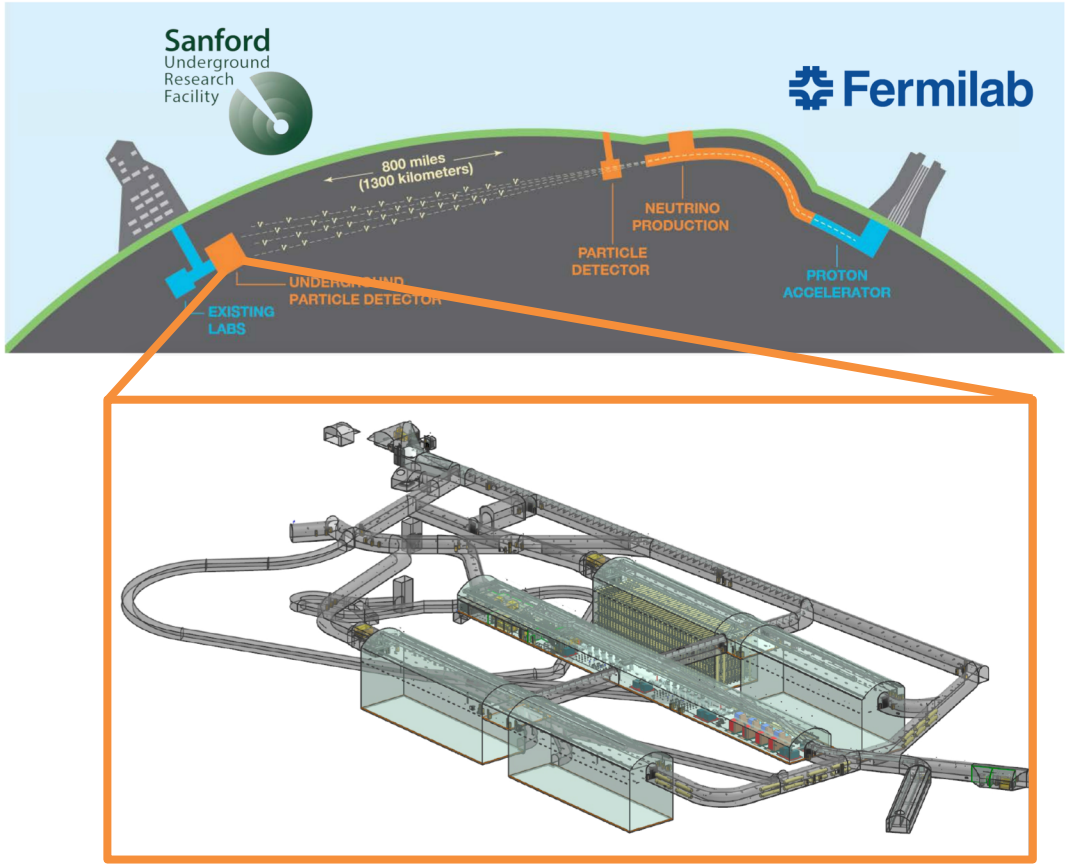


WHAT IS DUNE?

Deep Underground Neutrino Experiment

DEEP UNDERGROUND NEUTRINO EXPERIMENT

- Far Detectors (FD) are 800 miles from the neutrino beam source.
  - Four modules, each with 10,000 ton of liquid argon*.
- High power muon neutrino beam produced at Fermilab.
  - Can switch polarity to produce a muon antineutrino beam.
- Look for the appearance of electron (anti)neutrinos at the FD.
  - Measure CP-violation.



*active volume in the single phase module

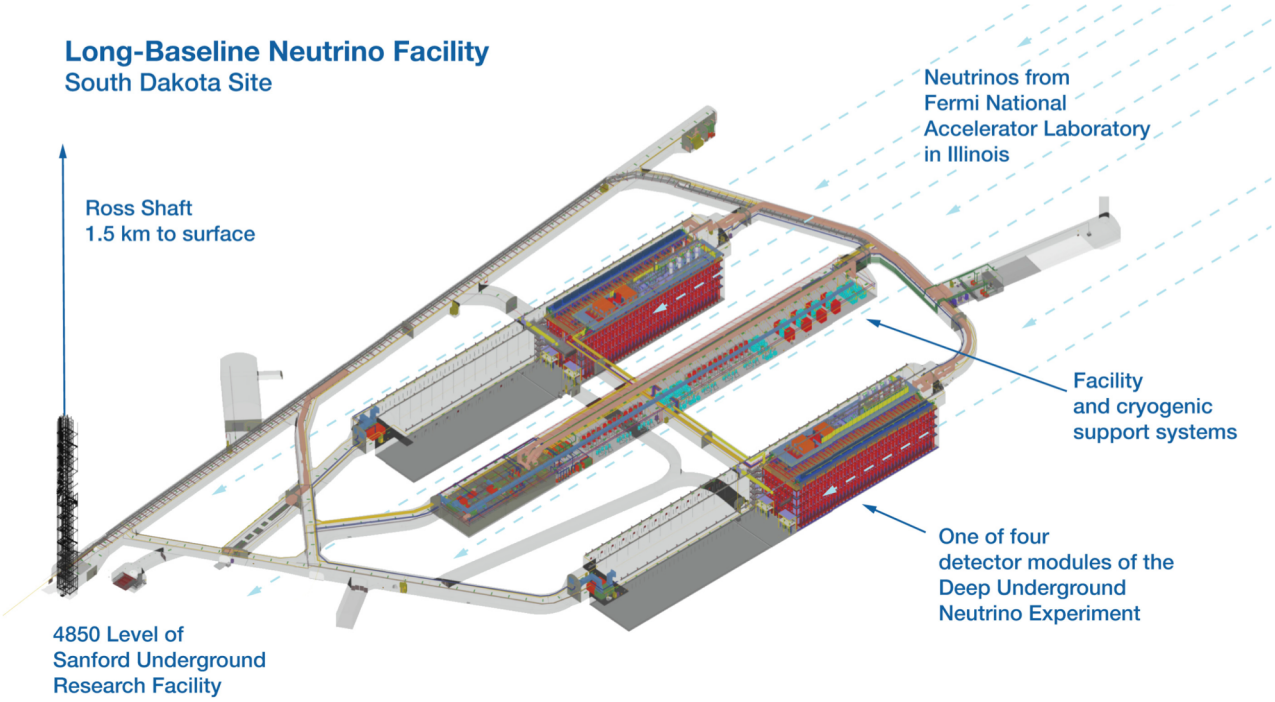
DUNE IN NUMBERS

Just for one Single-Phase modules (4 modules in total)

- **Number of Channels:** 384.000
- **TPC ADC Sampling:** 2 MHz
- **TPC ADC Dynamic Range:** 12 bits

- **Localized Event Record Window:** 5.4 ms
- **Full size of TPC Localized Event Record**: 6 GB

- **Extended Event Record Window:** 100 s
- **Full size of TPC Extended Event Record**: 115 TB



REAL-TIME DATA PROCESSING

How can we handle those numbers?

Our goal:

- Reduce the amount of data stored and reconstructed offline.
- We have experience in machine learning.
- Correctly applied it can help in the data selection or reconstruction.

REAL-TIME DATA PROCESSING

How can we handle those numbers?

Our constrains:

- Fast inference.
- Bandwidth operation.
- Long term operation.
- Power consumption.

REAL-TIME DATA PROCESSING

How can we handle those numbers?

Our constrains:

- Fast inference.
- Bandwidth operation.
- Long term operation.
- Power consumption.

FPGAS

MACHINE LEARNIG ON FPGAs

Two of the state-of-the-art approaches

- HLS4ML
  - Generate HLS code from neural network implementation.
  - No HDL programming.
  - Fast deployment using SDAccel.
- See Vladimir's talk
  https://indico.cern.ch/event/773049/contributions/3474297/

- Micron framework*
  - Inference engine optimized for ML.
  - No HDL programming.
  - Efficient use of memory bandwidth.
  - Direct from framework trained model to hardware.

*Formerly FWDNXT

MICRON FRAMEWORK

Direct deployment of neural networks on the inference engine

Micron framework[1]:

- No HDL programming.
- Natively supported neural networks.
- Most of the common layers are supported.
- Any framework that supports export to ONNX.
- Inference engine as an accelerator.

"Machine learning powers your world"

[1] https://fwdnxt.com/

INFERENCE ENGINE

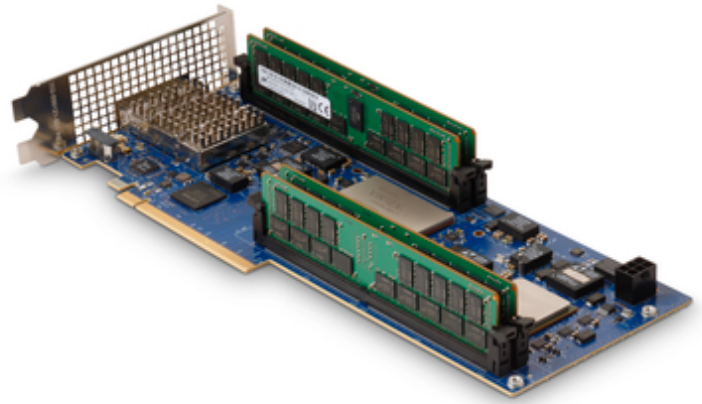
An FPGA ready for machine learning!

Micron Advanced Computing Solutions (ACS)

SB-852[1]:

- Xilinx Virtex Ultrascale+ UV9P.
- 64GB DDR4 SODIMM (up to 512GB).
- 2GB Hybrid Memory Cube.
- 2 QSFP transceiver connectors.
- PCIe x16 Gen3 to the host.

- High-bandwidth.
- Low-latency.



"The SB-852 is designed to deliver unprecedented levels of high-bandwidth and low-latency performance in the smallest possible footprint for advanced, high-performance applications."

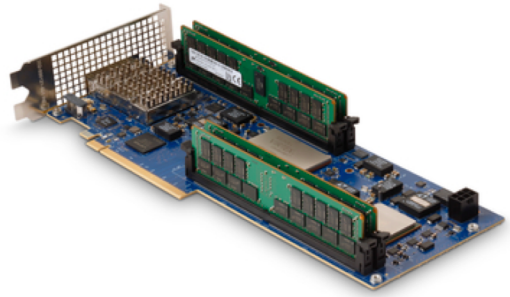
[1] https://www.micron.com/products/advanced-solutions/advanced-computing-solutions/hpc-single-board-accelerators/sb-852

MICRON FRAMEWORK

Inference engine + high performance compiler

Workflow:

1. Train your network.
2. Convert it into ONNX.
3. Compile it using the Micron framework.
4. Deploy into the inference engine.



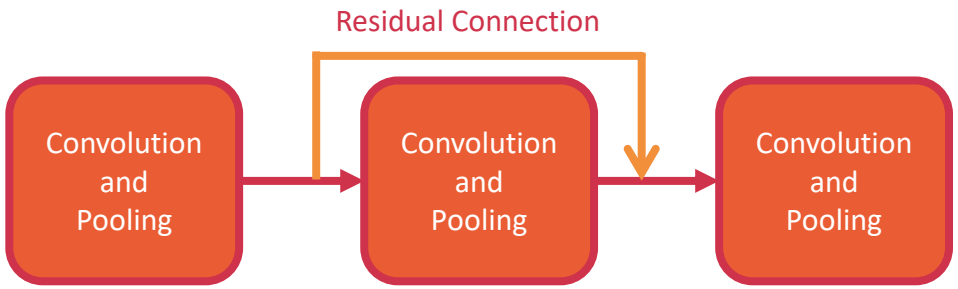
LET'S GIVE IT A TRY!

Testing the Micron framework and the inference engine

DUNE CVN

DUNE Convolutional Visual Network

- The DUNE Convolutional Visual Network[1] (CVN) is a CNN used for the neutrino identification task.
  - The DUNE CVN is inspired by the ResNet-18[2] architecture. It helps to preserve the fine-grained detail deeper in the network.



CVN ResNet-18
- Input 3x500x500
- Output 1x13
- Ops: 18.95 G-ops

ResNet-18 for ImageNet:
- Input 3x224x224
- Output 1x1000
- Ops 3.6 G-ops

[1]DUNE Collaboration. *Neutrino interaction classification with the DUNE Convolutional Visual Network*, under review within the DUNE collaboration.
[2]H. Kaiming et al., *Deep residual learning for image recognition*, CoRR, arXiv 1512.03385, 2015

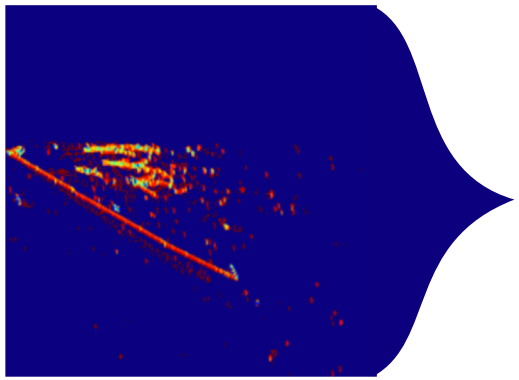
DUNE CVN ON FPGA

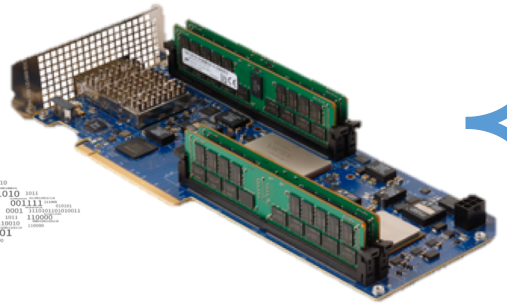
Testing the network on the inference engine

We tested the network over ~2M images from our DUNE CVN dataset.

Steps:

- Read the image :
  - Micron framework takes the image and send it to the inference engine main memory.
- Run the inference:
  - Ask the framework to run the inference and get the results.



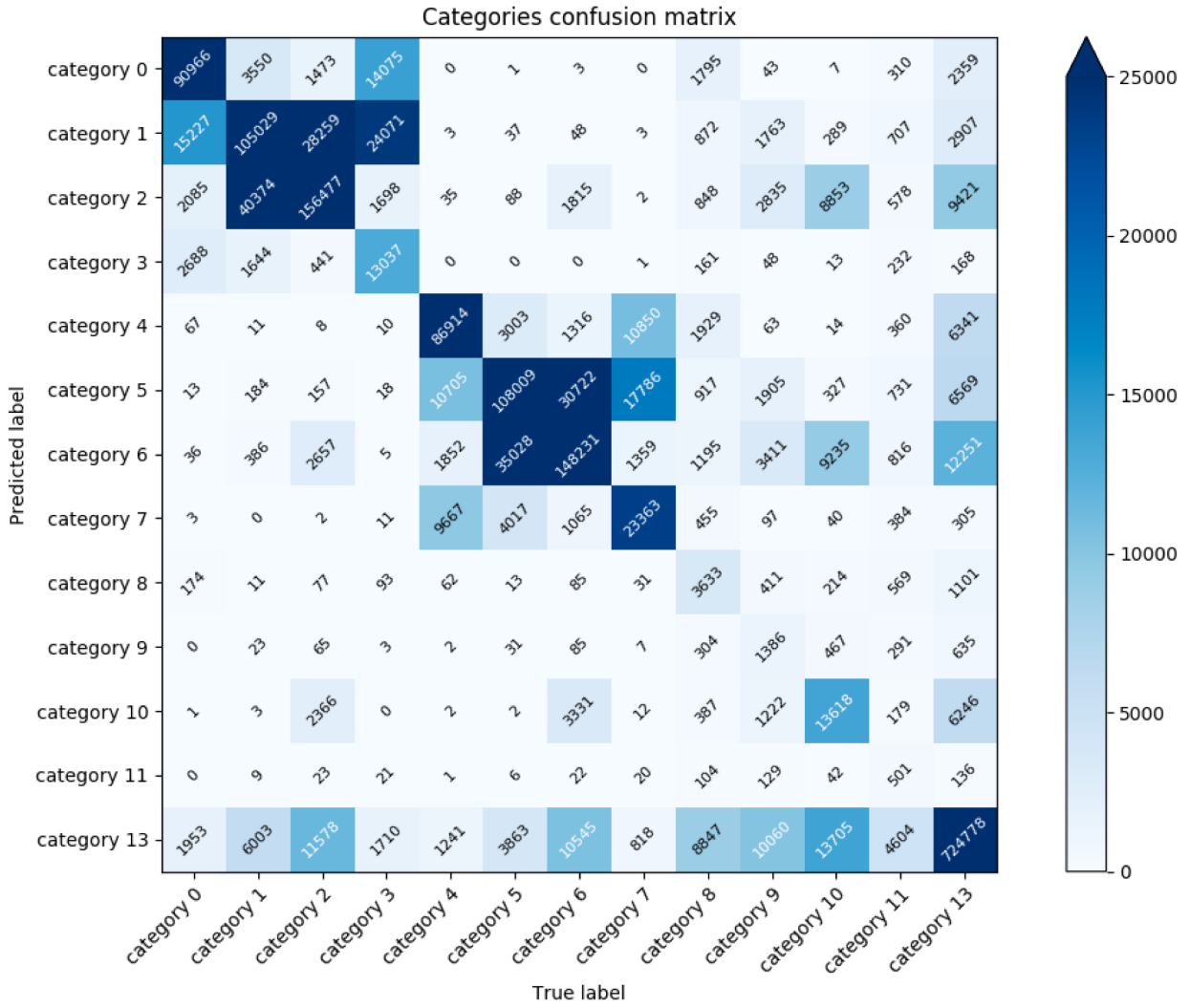


p(0) = 0.015
p(1) = 0.154
p(2) = 0.321
p(3) = 0.002
…
p(12) = 0.56

RESULTS

Classification report

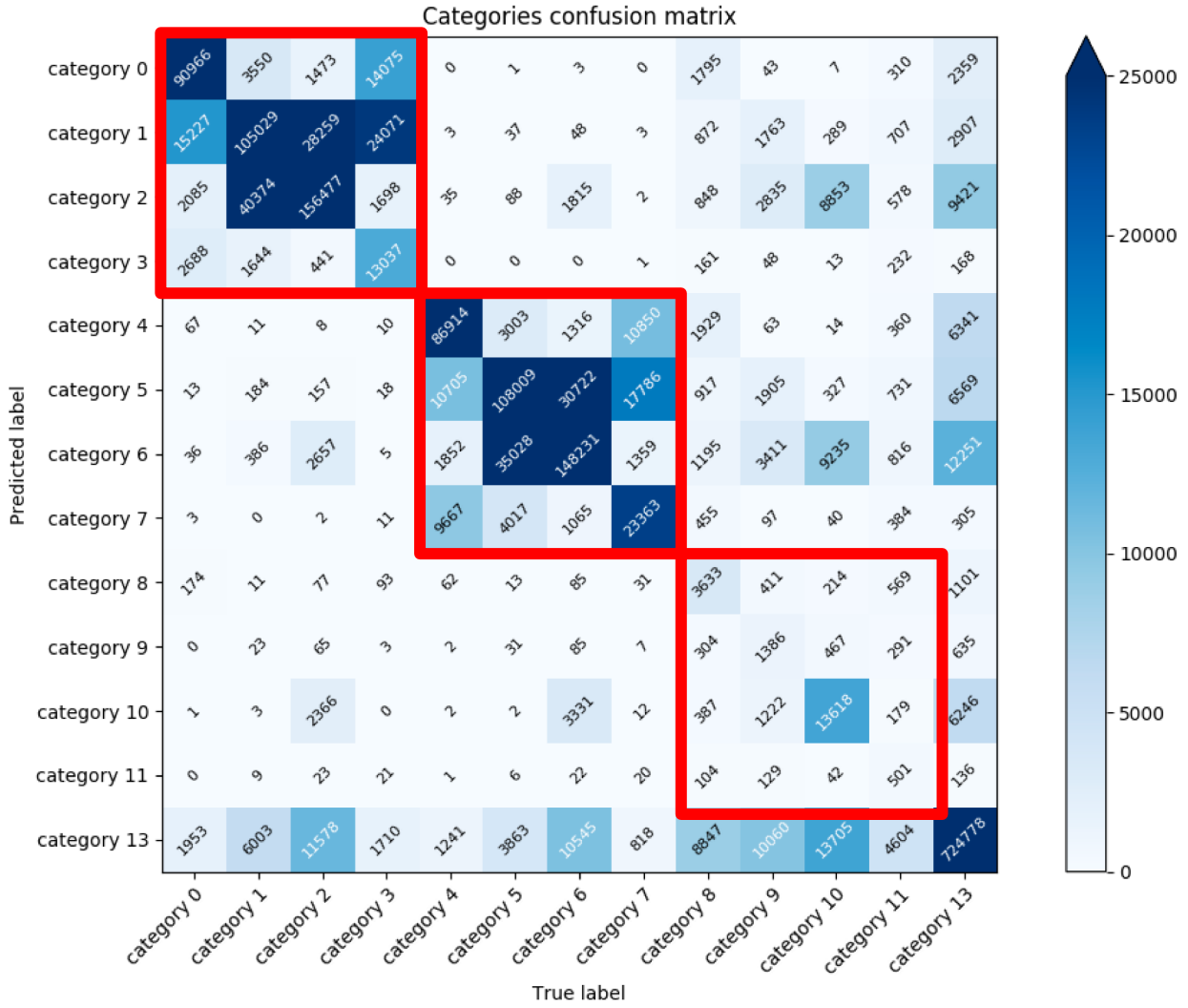
	precision	recall	f1-score	support
category 0	0.79	0.80	0.80	113213
category 1	0.59	0.67	0.62	157227
category 2	0.70	0.77	0.73	203583
category 3	0.71	0.24	0.36	54752
category 4	0.78	0.79	0.79	110484
category 5	0.61	0.70	0.65	154098
category 6	0.68	0.75	0.72	197268
category 7	0.59	0.43	0.50	54252
category 8	0.56	0.17	0.26	21447
category 9	0.42	0.06	0.10	23373
category 10	0.50	0.29	0.37	46824
category 11	0.49	0.05	0.09	10262
category 13	0.91	0.94	0.92	773217
accuracy			0.77	1920000
macro avg	0.64	0.51	0.53	1920000
weighted avg	0.76	0.77	0.76	1920000



RESULTS

Classification report

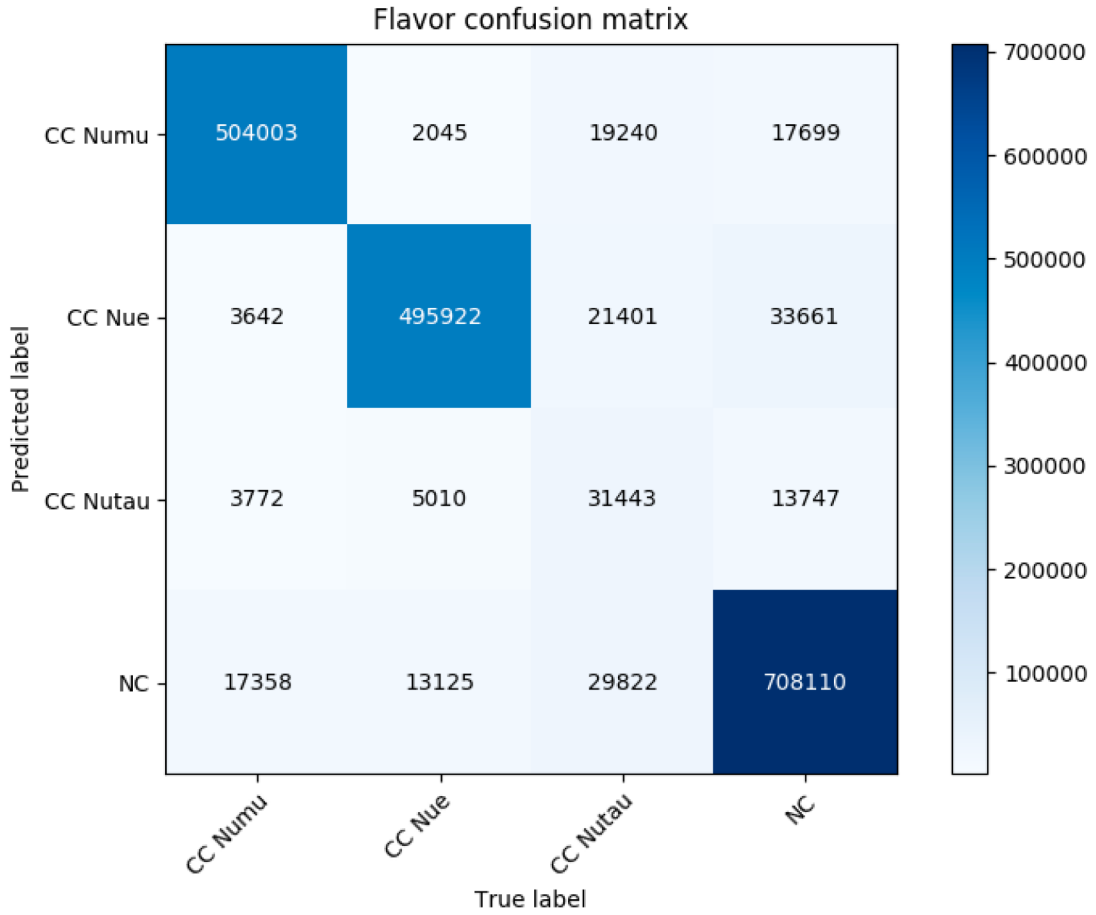
	precision	recall	f1-score	support
category 0	0.79	0.80	0.80	113213
category 1	0.59	0.67	0.62	157227
category 2	0.70	0.77	0.73	203583
category 3	0.71	0.24	0.36	54752
category 4	0.78	0.79	0.79	110484
category 5	0.61	0.70	0.65	154098
category 6	0.68	0.75	0.72	197268
category 7	0.59	0.43	0.50	54252
category 8	0.56	0.17	0.26	21447
category 9	0.42	0.06	0.10	23373
category 10	0.50	0.29	0.37	46824
category 11	0.49	0.05	0.09	10262
category 13	0.91	0.94	0.92	773217
accuracy			0.77	1920000
macro avg	0.64	0.51	0.53	1920000
weighted avg	0.76	0.77	0.76	1920000



RESULTS

Flavor report

	precision	recall	f1-score	support
CC Numu	0.93	0.95	0.94	528775
CC Nue	0.89	0.96	0.93	516102
CC Nutau	0.58	0.31	0.40	101906
NC	0.92	0.92	0.92	773217
accuracy			0.91	1920000
macro avg	0.83	0.78	0.80	1920000
weighted avg	0.90	0.91	0.90	1920000



PERFORMANCE

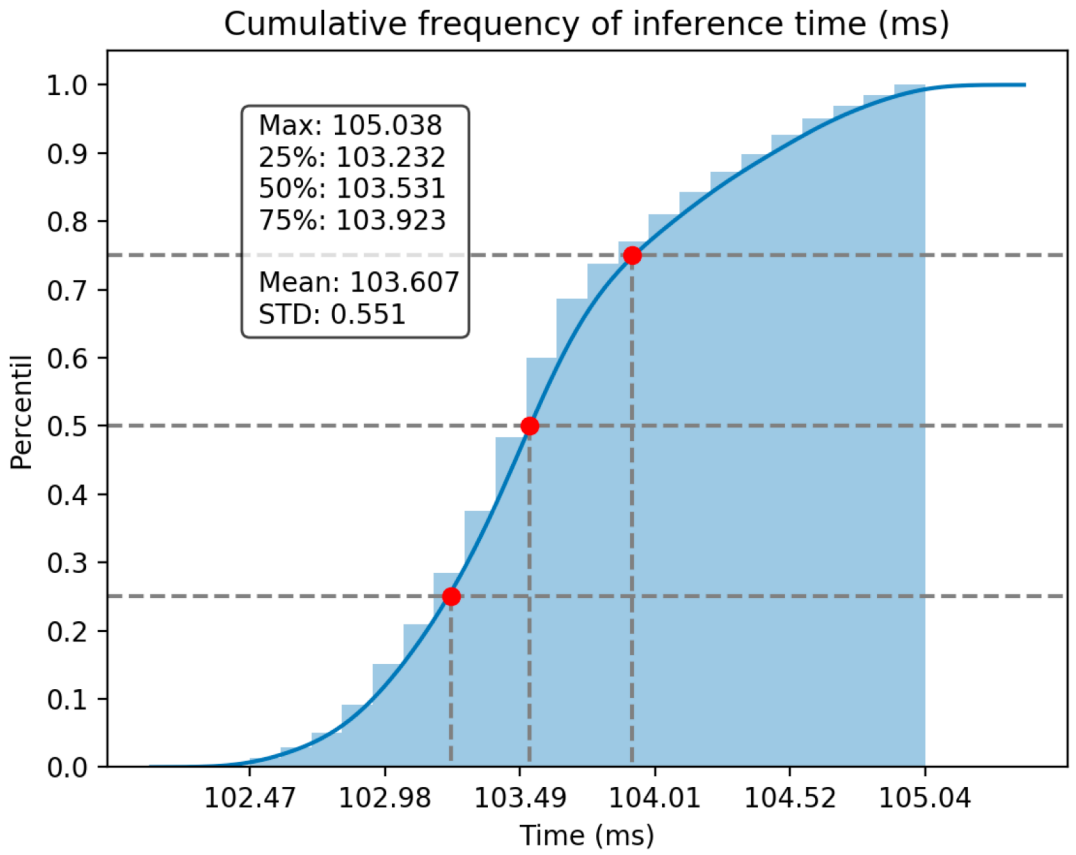
ResNet-18

- Input 3x500x500
- Ops: 18950766592 (18.95 G-ops)

Inference Engine:

- SB-852 4th Gen
- MACs: 512
- Clock frequency: 250 MHz
- Ops per MAC: 2

$$Inference\ time\ = \frac{OPS_{ResNet18} \cdot Clock_{Cycle}}{MACs\ \cdot OPS_{MAC}} = 74.0264\ ms$$



PERFORMANCE

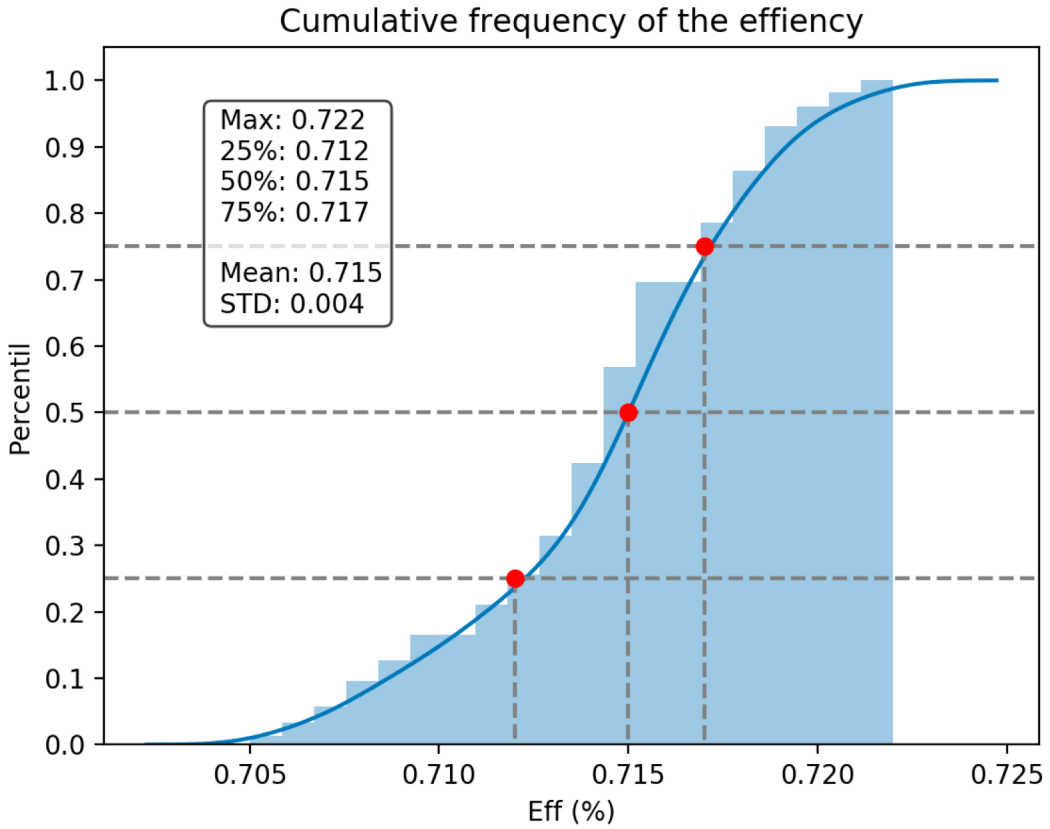
ResNet-18

- Input 3x500x500
- Ops: 18950766592 (18.95 G-ops)

Inference Engine:

- SB-852 4th Gen
- MACs: 512
- Clock frequency: 250 MHz
- Ops per MAC: 2

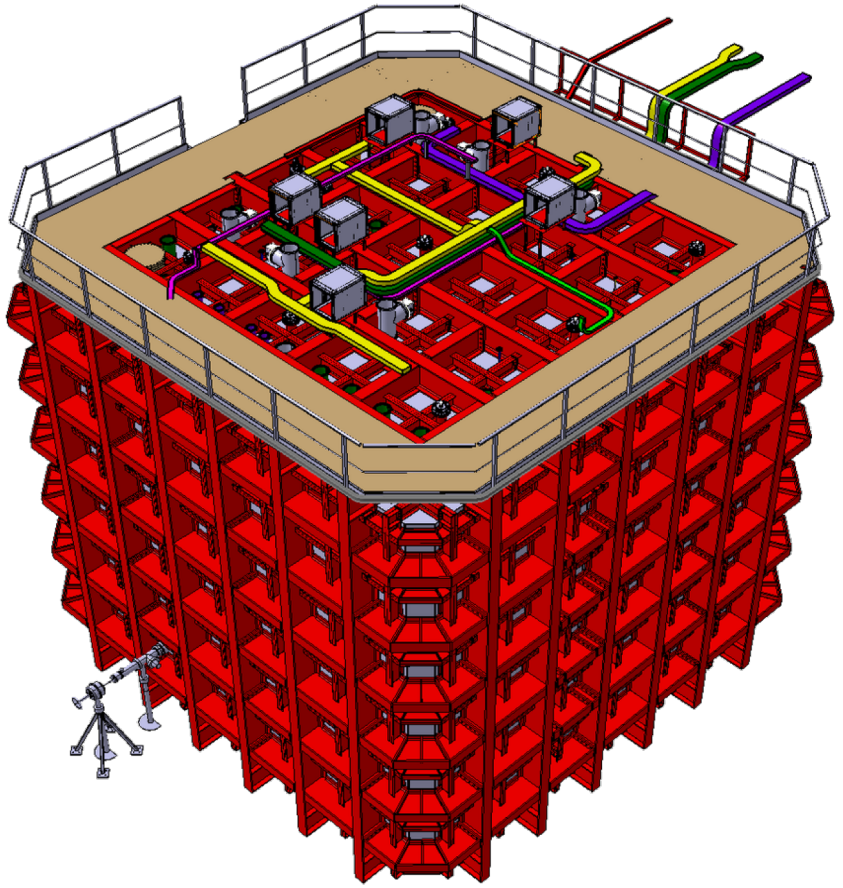
$$Inference\ time\ = \frac{OPS_{ResNet18} \cdot Clock_{Cycle}}{MACs\ \cdot OPS_{MAC}} = 74.0264\ ms$$



FUTURE PLANS

Data selection and trigger generation

- Implement the inference engine into protoDUNE-SP DAQ.
- Online hit-findings approaches.
- Use the hits information and CNN to identify **trigger candidates**.
- See how far we could go.



See Roland's talk: https://indico.cern.ch/event/773049/contributions/3474337/

SUMMARY

That's all folks!

- Looked for different fast inference solutions for ML → FPGAs.
- Micron framework → ML framework + Inference Engine.
- We tested it using the current DUNE CVN.
- Really good results.
- Now we're moving to real-time application.
- Integrate a CNN in protoDUNE-SP DAQ chain to get self-triggered data.

THANK YOU!

BACKUP

INFERENCE TIME ON CPU

1 Core	Time (ms)
	770.095
	771.5236
	771.4623
	772.3615
AVG:	771.3606

3 Cores	Time (ms)
	310.6509
	310.6518
	311.0137
	311.6450
AVG:	310.9904

2 Cores	Time (ms)
	449.3526
	446.7231
	448.3936
	481.1092
AVG:	456.3946

48 Cores	Time (ms)
	50.7780
	50.4031
	50.3375
	50.5165
AVG:	50.5088

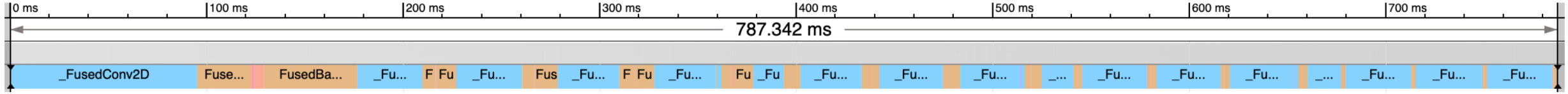
```
Model name:            Intel(R) Xeon(R) Silver 4116 (x2)
Frequency:             2.10GHz
CPU(s):                48
On-line CPU(s) list:   0-47
Thread(s) per core:    2
Socket(s):             2
NUMA node(s):          2
NUMA node0 CPU(s):     0-11,24-35
NUMA node1 CPU(s):     12-23,36-47
Price*:                1'659.98$ (2 x 829.99$)
TDP**:                 85W
```

*: From Intel website
**: Thermal Design Power (TDP) represents the average power, in watts, the processor dissipates when operating at Base Frequency with all cores active under an Intel-defined, high-complexity workload.

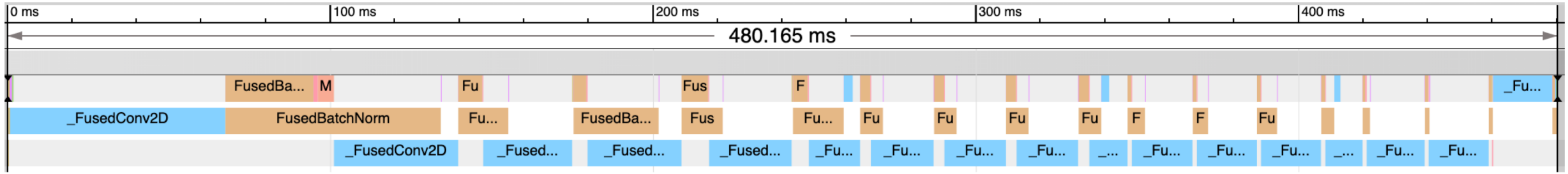
INFERENCE TIME ON CPU

Profiling results

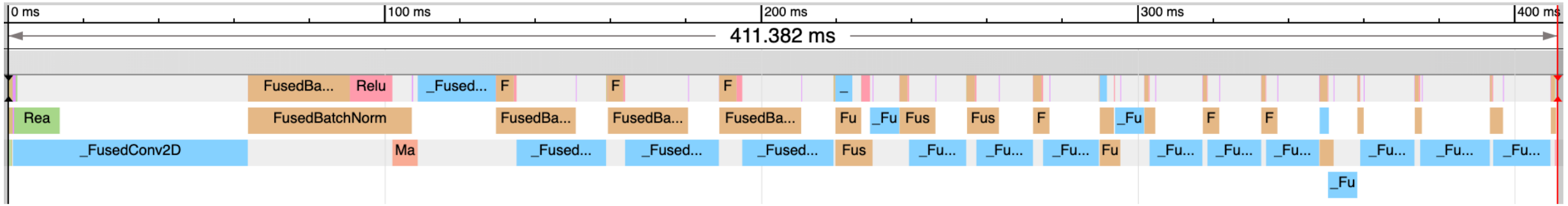
One CPU



Two CPU



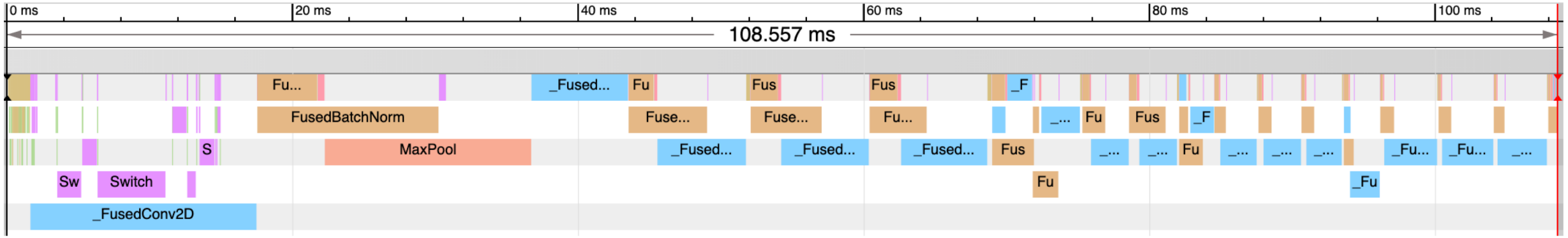
Three CPUs



INFERENCE TIME ON CPU

Profiling results

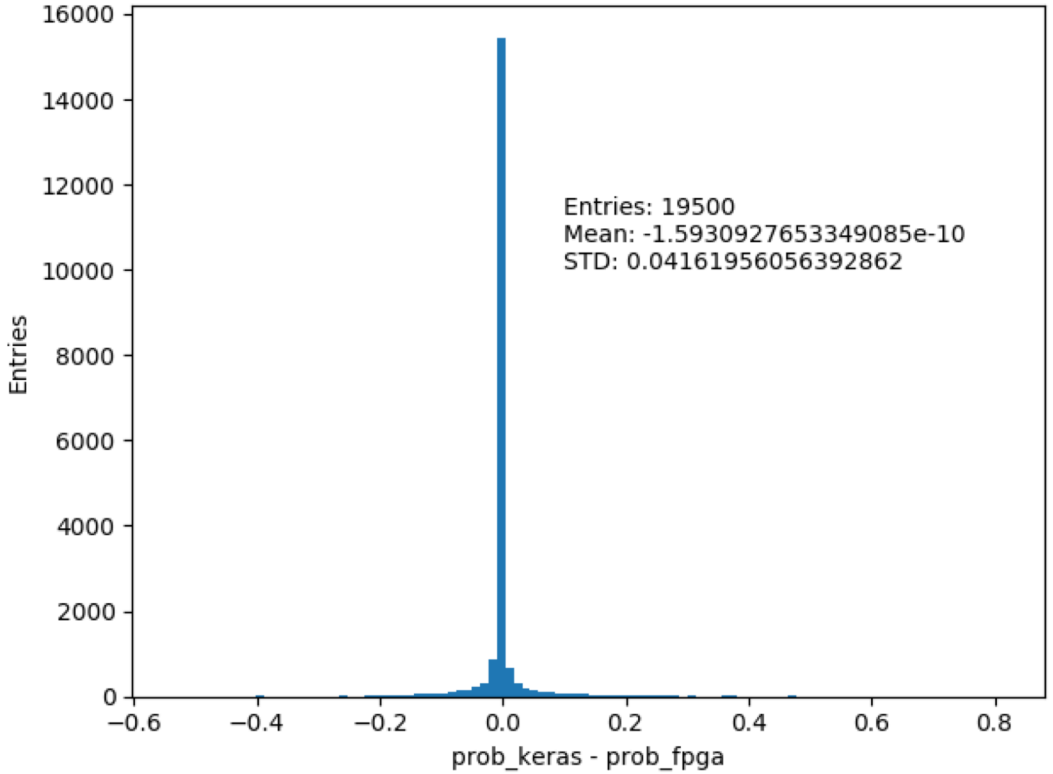
Entire server 48 CPUs:

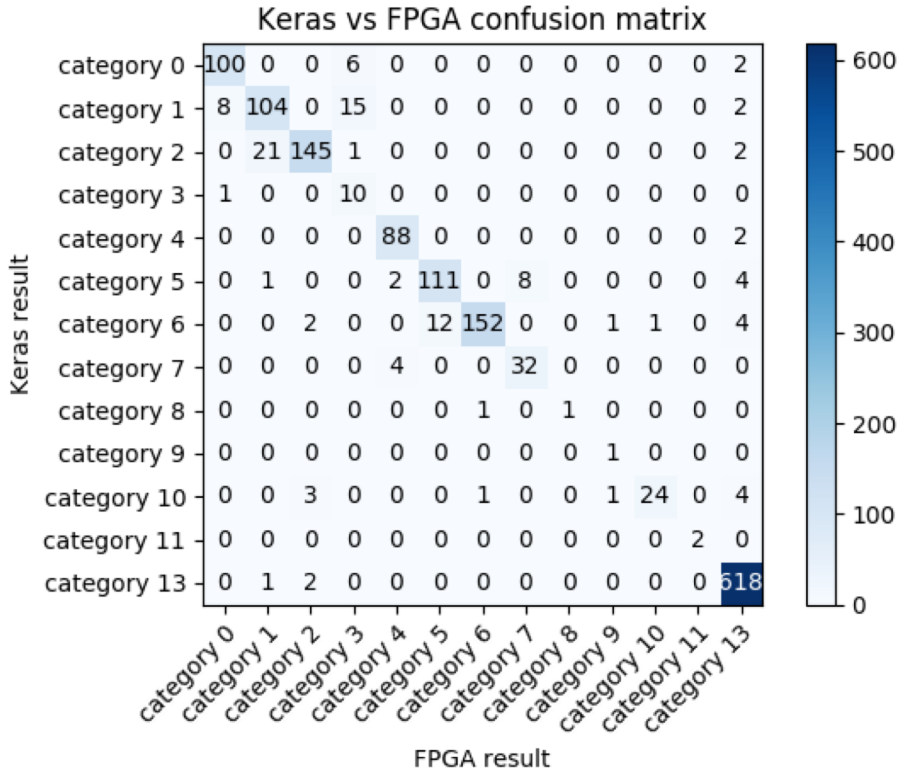



```
Model name:            Intel(R) Xeon(R) Silver 4116
Frequency:             2.10GHz
CPU(s):                48
On-line CPU(s) list:   0-47
Thread(s) per core:    2
Socket(s):             2
NUMA node(s):          2
NUMA node0 CPU(s):     0-11,24-35
NUMA node1 CPU(s):     12-23,36-47
```

GPU-FPGA RESULTS COMPARISON

How good our FPGA behaves





COST PER INFERENCE

Money is all that matters

$$\frac{Cost}{Inference} = \frac{Time}{Inference} \cdot TDP \cdot Cost\ of\ Enery = K \cdot Cost\ of\ Energy$$

Device	Inference Time (ms)	TDP (W)
GPU (Nvidia Tesla K80)	37.7344	300
CPU (2x Intel Xeon Silver 4116)	50.5088	170*
FPGA (Micron SB-852)	103.6074	70

*85W (x2)

	GPU	CPU	FPGA
K factor	11.3203	8.5865	7.2550