

# Contributions to the ePIC DAQ Project

TAKU GUNJI

CENTER FOR NUCLEAR STUDY

THE UNIVERSITY OF TOKYO

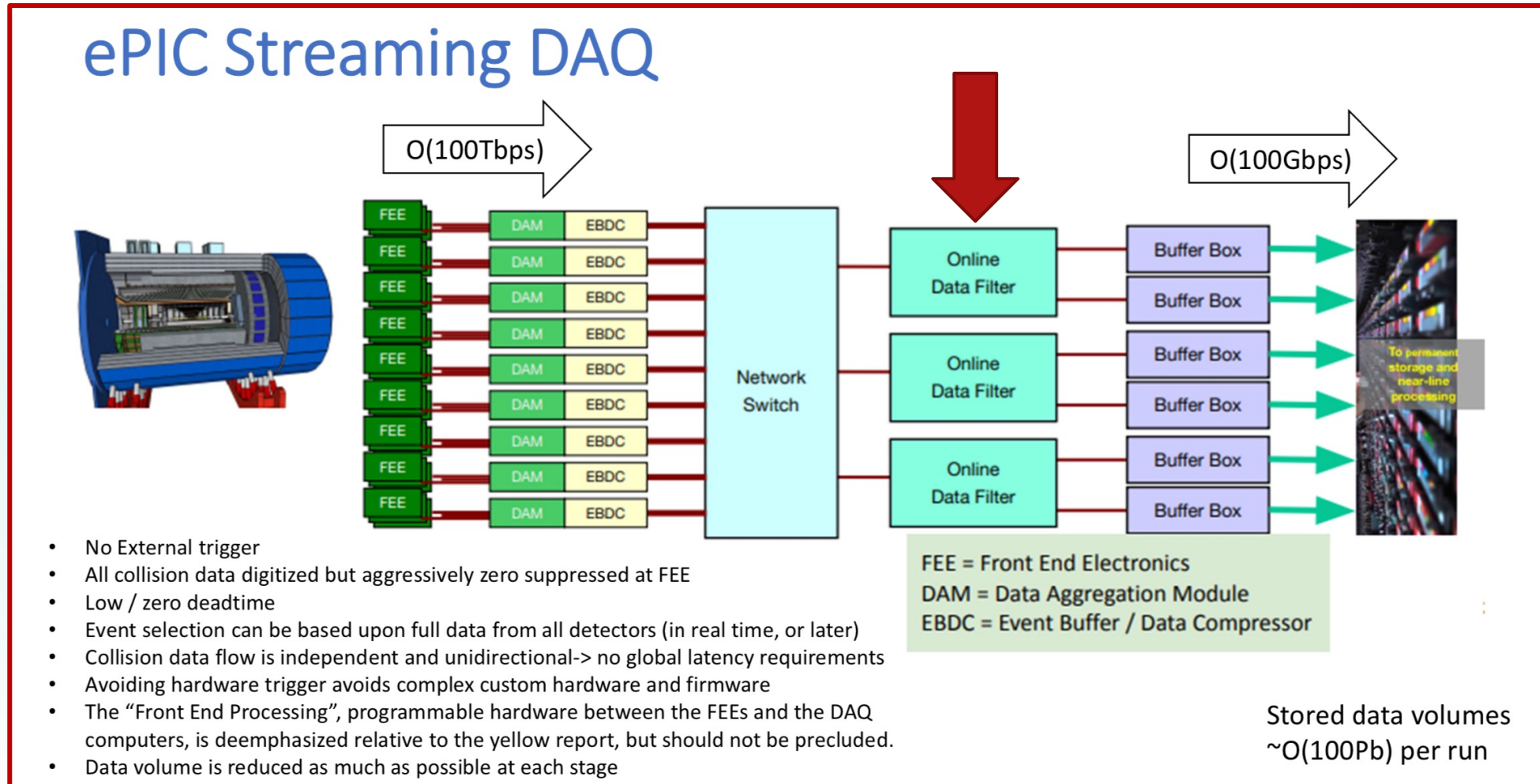


# Outline

- ▶ Free Streaming Readout
- ▶ ALICE O2
- ▶ Some (small) activities
- ▶ SPADI-Alliance
- ▶ Summary

# Free Streaming Readout

- ▶ Will be a future standard DAQ system



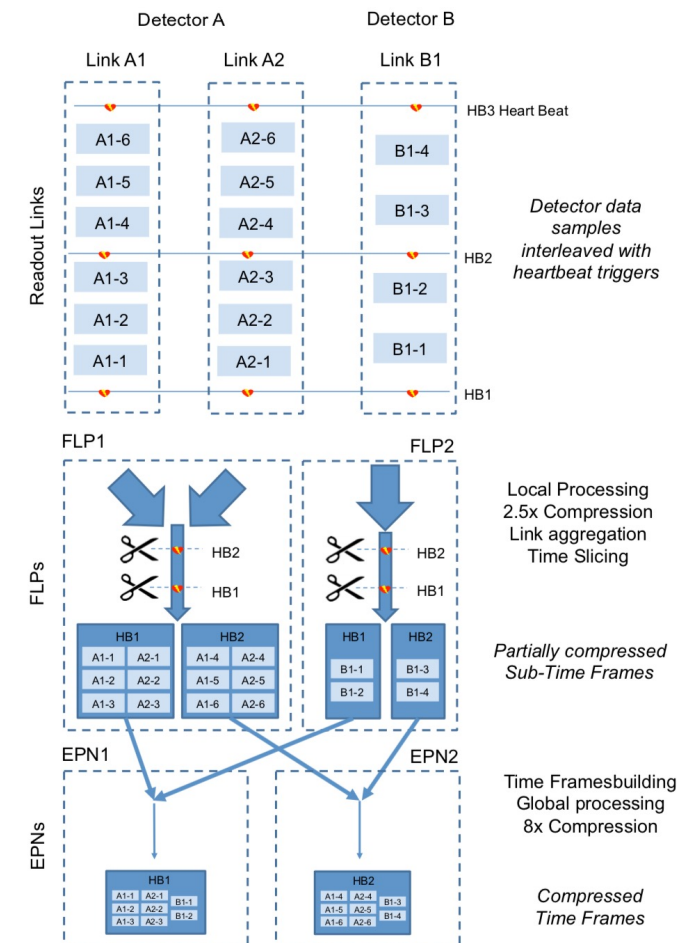
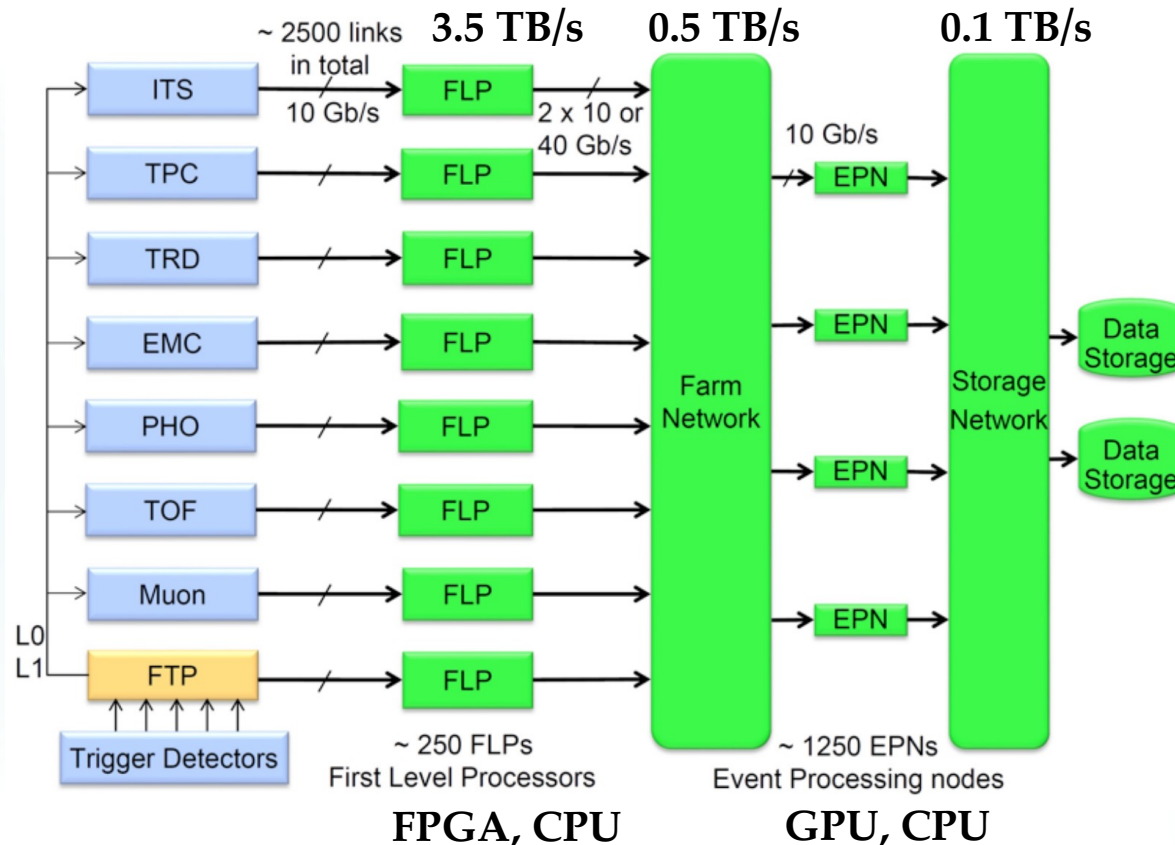
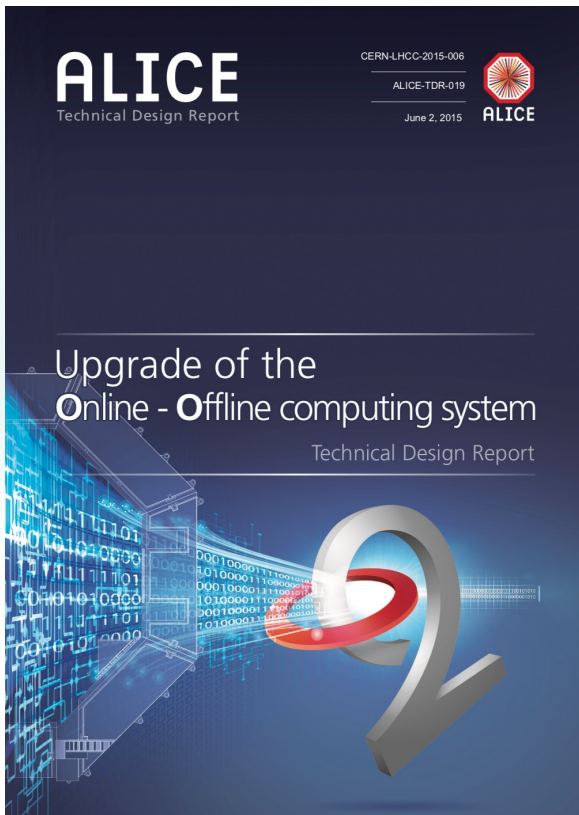
# ALICE O<sup>2</sup> system

50kHz Pb-Pb collisions

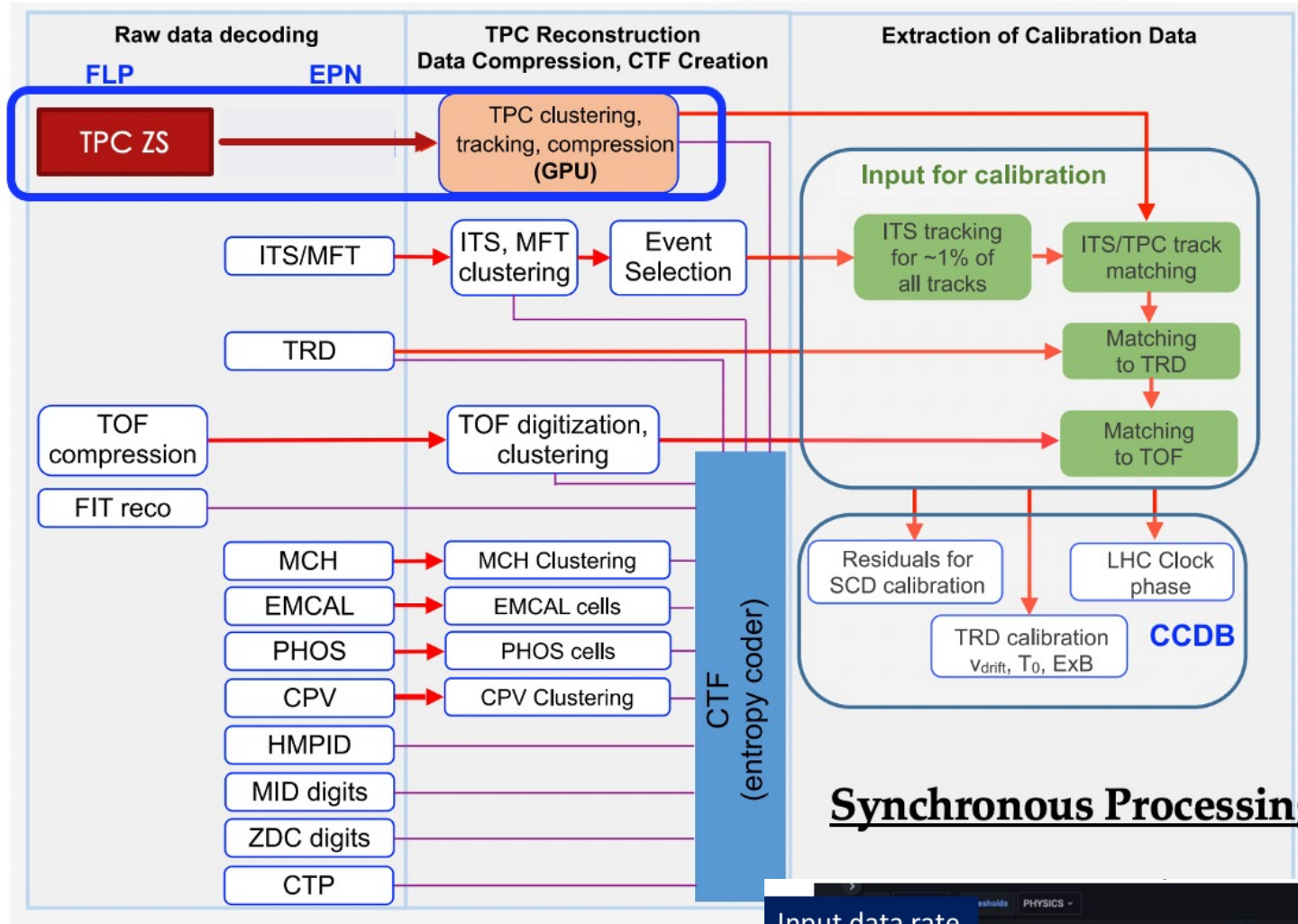
TPC moved from triggered readout (gating grid) to continuous readout with GEMs

ALICE had experiences on hardware acceleration (GPUs) in HLT during Run2

→ New Free streaming DAQ (O<sup>2</sup>) proposed in 2015



# Online processing in ALICE O<sup>2</sup>



8 GPU/EPN, 250 EPNs



**Synchronous Processing**

Input data rate										Compressed Data	Raw Data
PHYSICS -											
FLP		EPN						CTF writer		EOS	
Readout	StfBuilder	DPL In	DPL Out	StfSender In	StfSender Out	TfBuilder In	TfBuilder Out	DPL In	CTF writer	EOS	
95.6 GB/s	95.0 GB/s	93.1 GB/s	91.9 GB/s	94.0 GB/s	109 GB/s	70.9 GB/s	85.4 GB/s	93.5 GB/s	868 MB/s	90.4 GB/s	

# Online processing performance

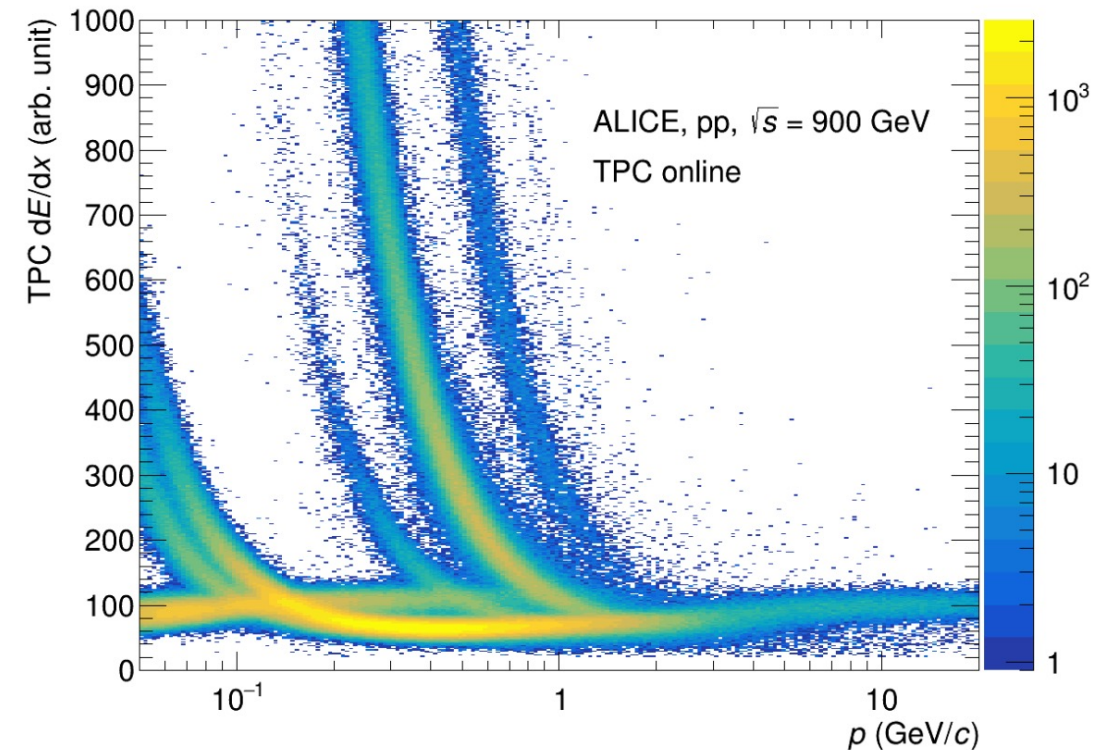
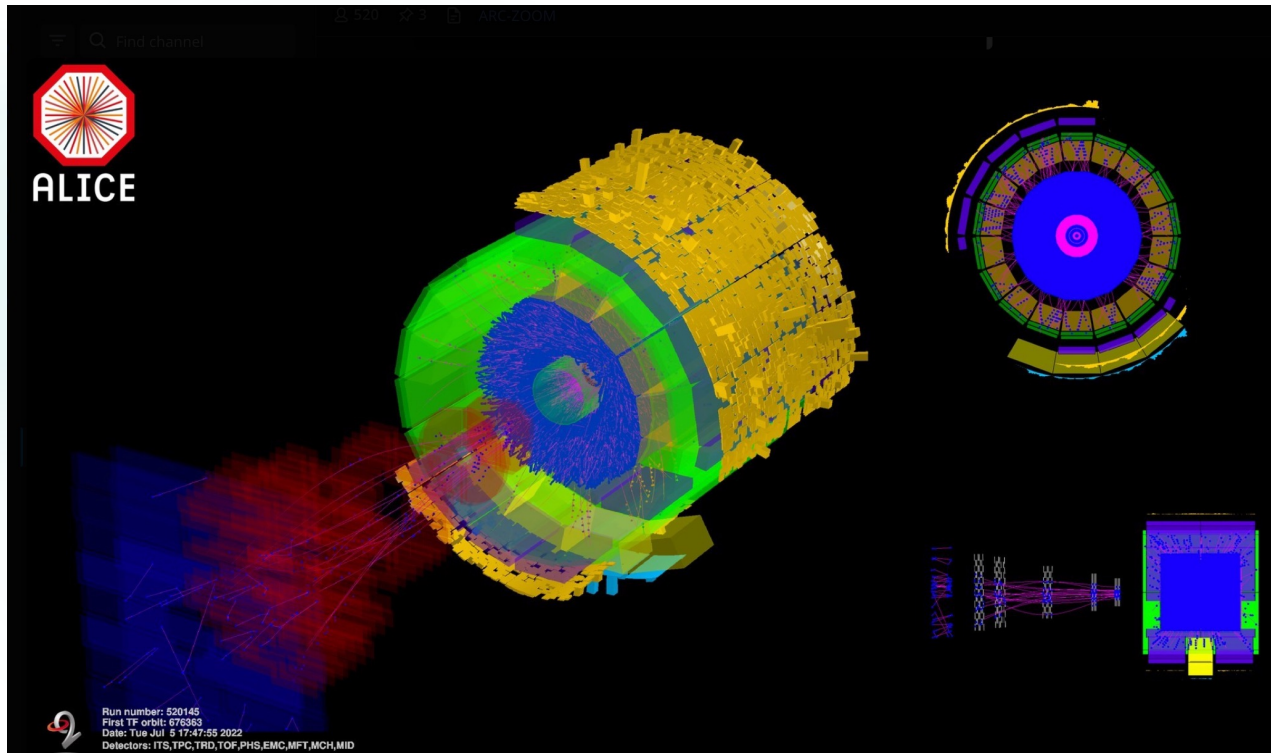
6

## ► Online clustering, tracking, PID

All are done **ONLINE**

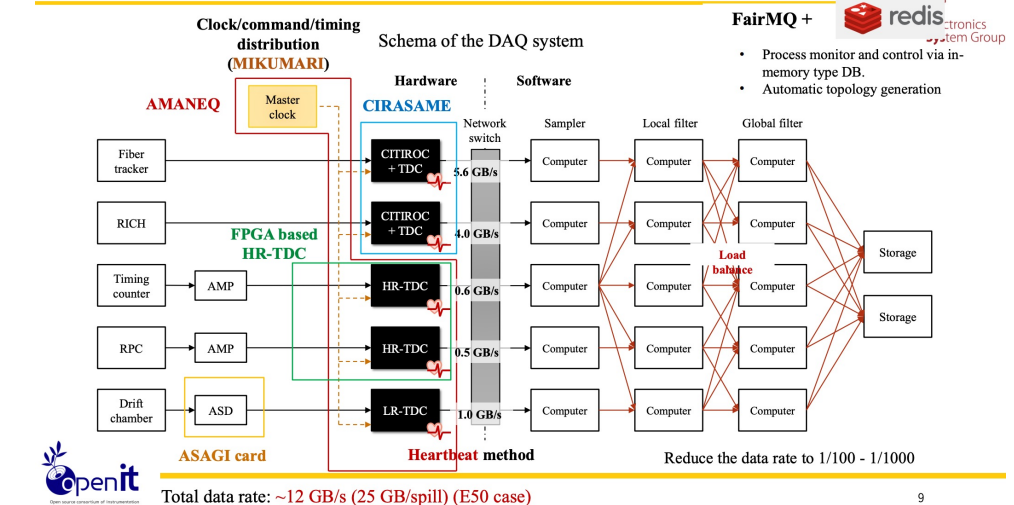
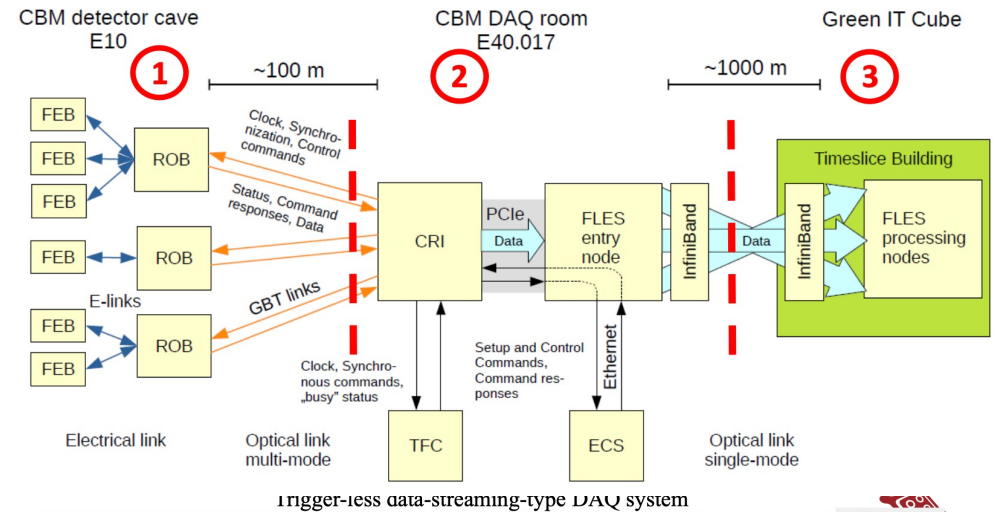
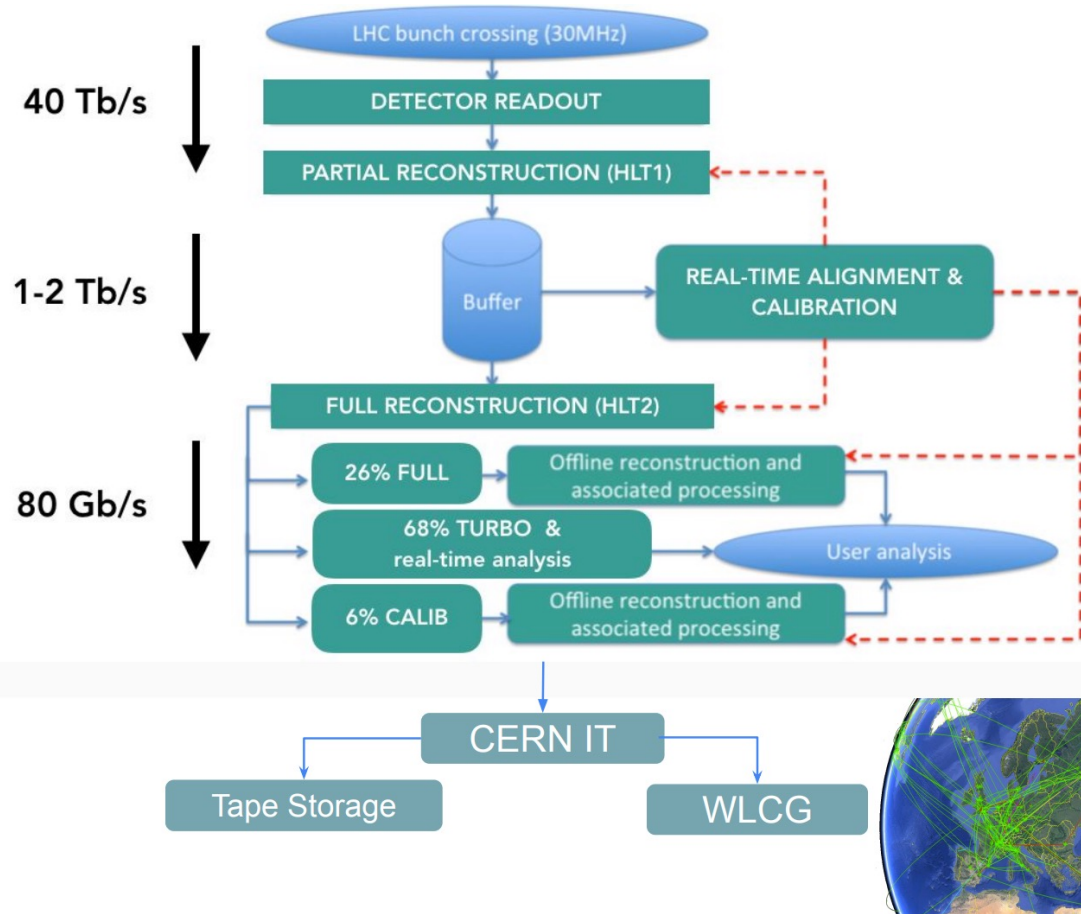
Event display after online reconstruction

1 TF = 11.5 ms, 10 kHz IR → ~100 collisions



# Will be used in many experiments

▶ ALICE, LHCb, sPHENIX, CBM, J-lab, J-PARC, etc...

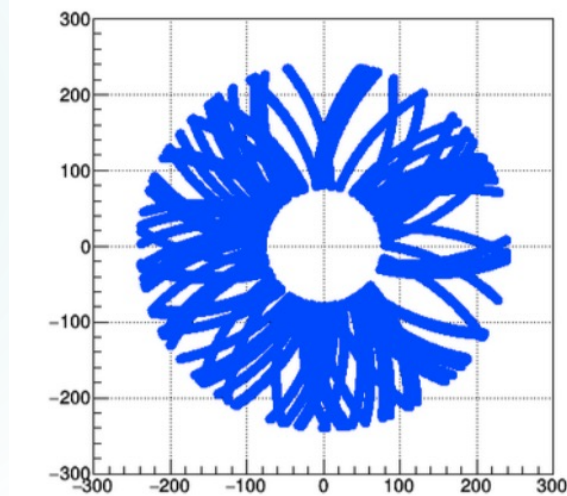
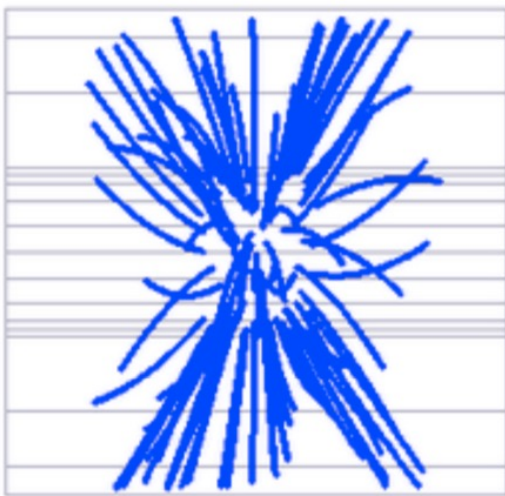


# My activities: GPU processing

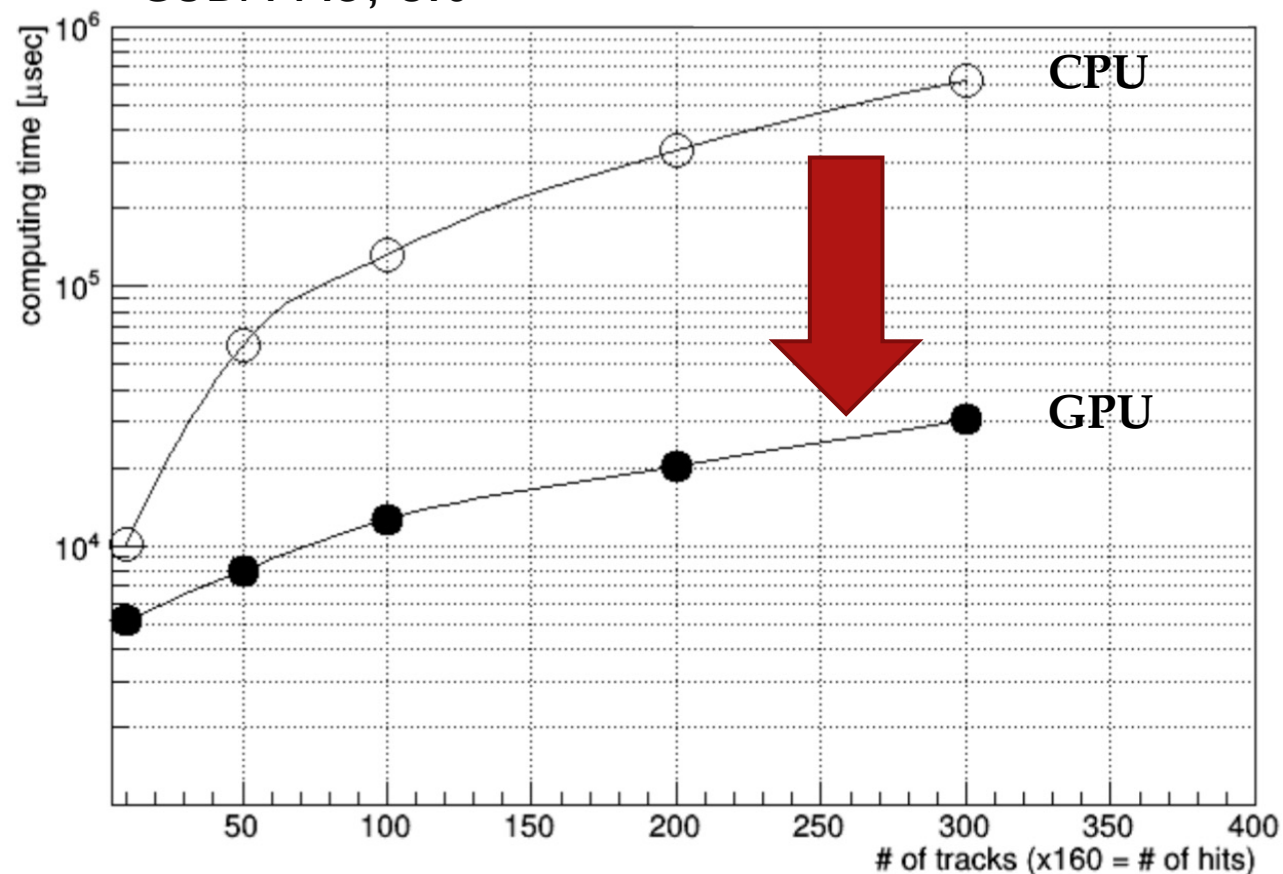
- ▶ Kalman filter tracking
- ▶ ML for pile-up identification

These works were done in 2016.

## TPC Tracking



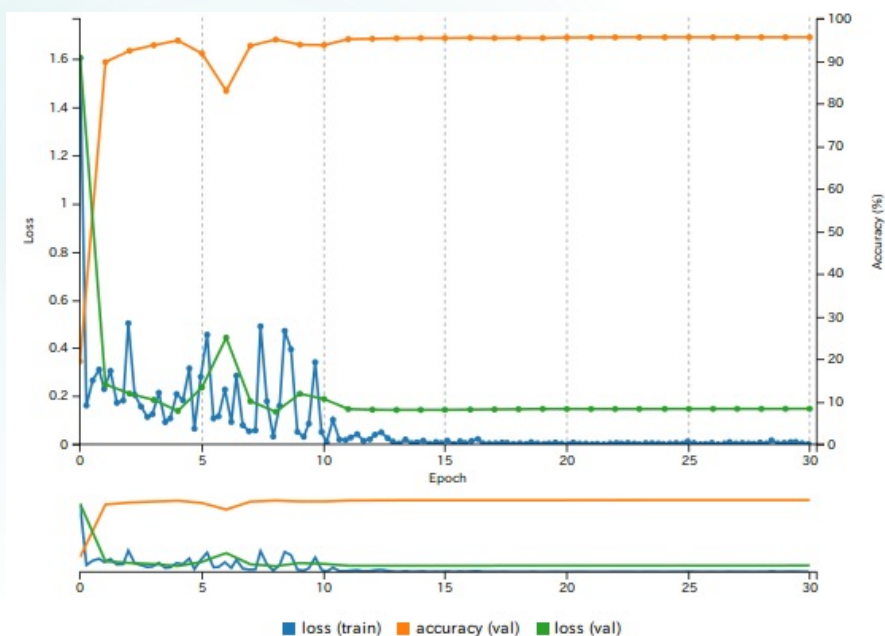
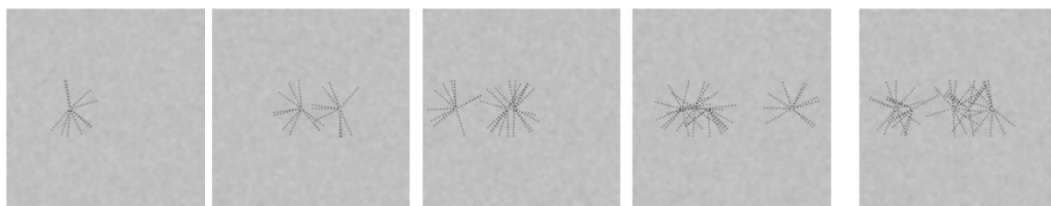
GPU = NVIDIA GTX970  
CUDA 7.5, 8.0





# My activities: GPU processing

- ▶ Kalman filter in GPU
- ▶ ML for pile-up identification



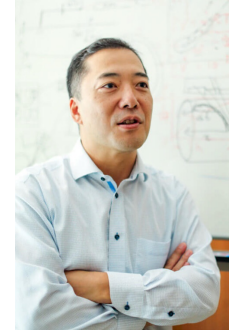
LeNET(vtxtmp) Image Classification Model

Path	Top predictions				
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/1/0000.png	1 100.0%	2 0.0%	3 0.0%	4 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/2/0000.png	2 100.0%	1 0.0%	3 0.0%	4 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/3/0000.png	3 99.99%	4 0.01%	2 0.0%	1 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/4/0000.png	4 99.99%	5 0.01%	3 0.0%	2 0.0%	1 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/5/0000.png	5 100.0%	4 0.0%	1 0.0%	3 0.0%	2 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/1/0001.png	1 100.0%	2 0.0%	3 0.0%	4 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/2/0001.png	2 100.0%	1 0.0%	3 0.0%	4 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/3/0001.png	3 99.92%	2 0.08%	4 0.0%	1 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/4/0001.png	4 99.8%	5 0.2%	3 0.0%	2 0.0%	1 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/5/0001.png	5 100.0%	4 0.0%	3 0.0%	1 0.0%	2 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/1/0002.png	1 100.0%	2 0.0%	3 0.0%	4 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/2/0002.png	2 100.0%	3 0.0%	1 0.0%	4 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/3/0002.png	3 99.92%	2 0.08%	4 0.0%	1 0.0%	5 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/4/0002.png	4 54.47%	5 45.53%	3 0.0%	1 0.0%	2 0.0%
/home/gunji/NVIDIA_CUDA-7.5_Samples/deep_learning/tmp/vtxtmp/test/5/0002.png	5 99.42%	4 0.58%	3 0.0%	1 0.0%	2 0.0%

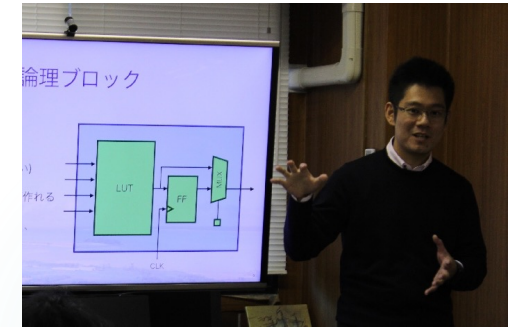
# TPC cluster finder in FPGA

## ▶ ALICE TPC clustering on FPGA

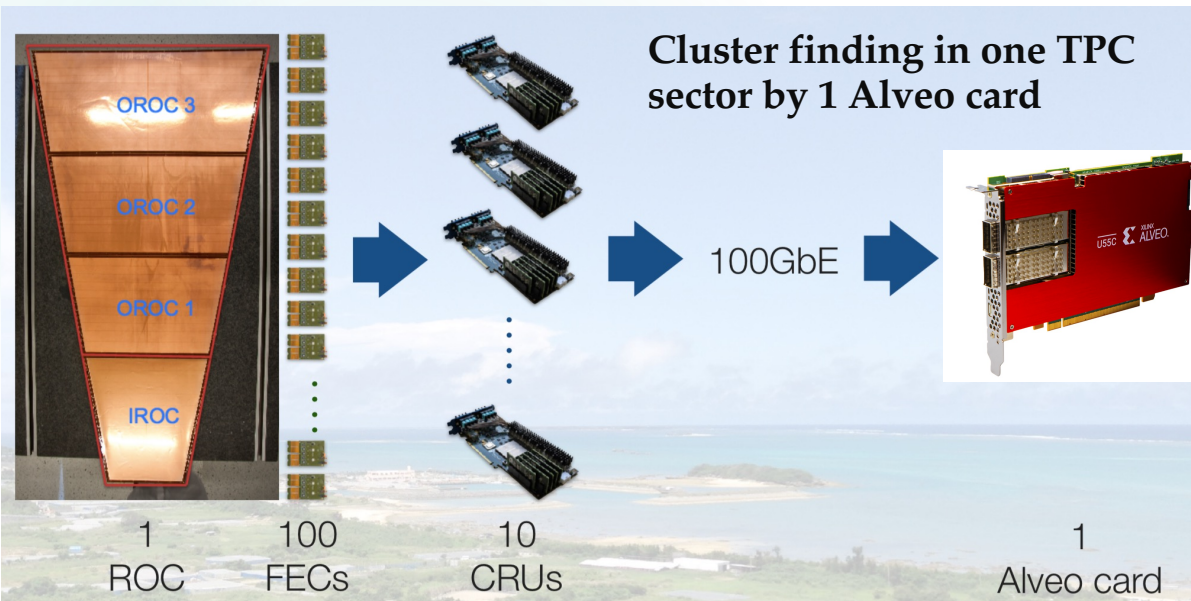
- ▶ Xilinx Alveo u55c, u280
- ▶ High Level Synthesis (C++)



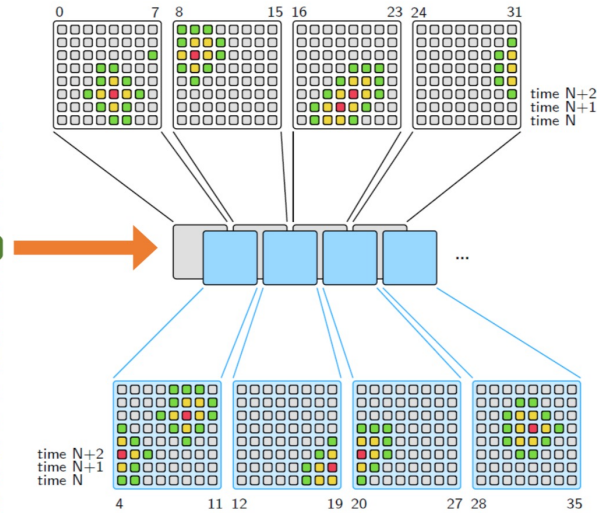
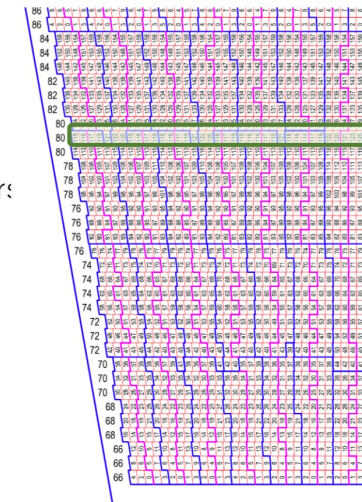
Ken Oyama (NiAS)



Yasunori Osana (Rukyū -> Komamoto)



- The **heaviest processing** in our FPGA
- find local maxima in pad-timebin 2D space
- processor modules
  - scan rectangular regions
  - optimize size v.s. number of processors:
  - available clock cycle is the boundary condition

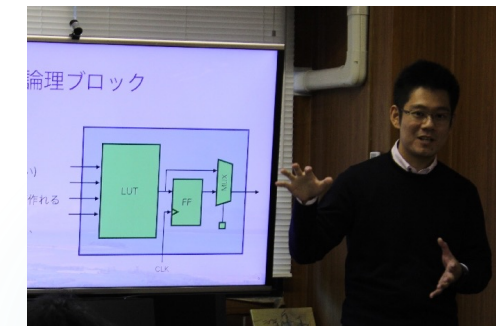
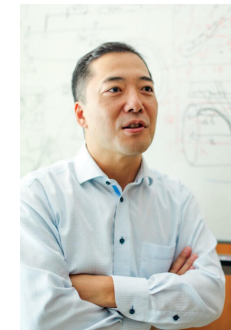


# TPC cluster finder in FPGA

## ▶ ALICE TPC clustering on FPGA

▶ Xilinx Alveo u55c, u280

▶ High Level Synthesis (C++)

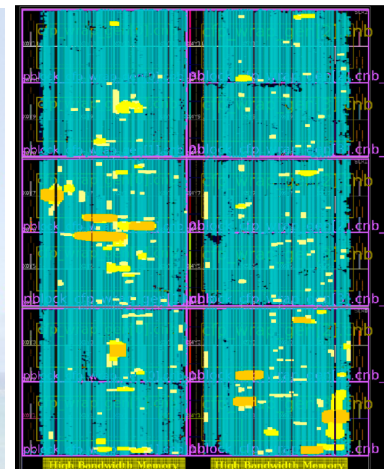
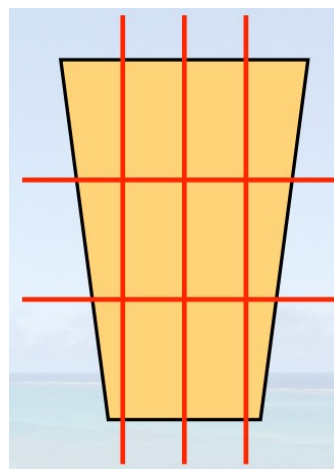


Ken Oyama (NiAS)    Yasunori Osana (Rukyū -> Komamoto)

```

1 #include "cfinder.h"
2 #include <fstream>
3
4 void cfinder(hls::stream<bus_t>& in1,
5             hls::stream<bus_t>& in2,
6             hls::stream<bus_t>& in3,
7             hls::stream<bus_t>& in4,
8             hls::stream<bus_t>& in5,
9             hls::stream<cluster_bus_t>&
10              out){
11     // #pragma HLS DATA_PACK variable=out
12     #pragma HLS INTERFACE axis register both
13     port=in1,in2,in3,in4,in5
14
15     static int tb=0;
16     tb++;
17
18     row_loop:
19     for (int row=0; row<ROWS; row++){
20         bus_t in_val[5];
21         #pragma HLS ARRAY_PARTITION
22         variable=in_val dim=1 complete
23         // #pragma HLS ARRAY_PARTITION variable=in_val
24         // complete dim=1
25         in_val[0] = in1.read();
26         in_val[1] = in2.read();
27         in_val[2] = in3.read();
28         in_val[3] = in4.read();
29         in_val[4] = in5.read();
30
31         pad_t buf[TIMEBINS][COLS] = {};
32         #pragma HLS ARRAY_PARTITION variable=buf
33         complete dim=1
34         #pragma HLS ARRAY_PARTITION variable=buf
35         complete dim=2
36
37         for (int t=0; t<TIMEBINS; t++){
38             for (int c=0; c<COLS; c++){
39                 buf[t][c] =
40                 in_val[0].range(c*10+9, c*10);
41             }
42         }
43         // Calculate cluster properties +
44         // Peak detection
45         // Q_max
46
47         cluster_bus_t tmp = {};
48         cluster clusters(COLS) = {};
49         #pragma HLS ARRAY_PARTITION
50         variable=clusters complete dim=1
51         for (int c=LR_CELLS; c<COLS-LR_CELLS;
52              c++){ // 2 to COLS-22
53             cluster cl;
54             // Q_tot
55             cl.qtot = 0; // pad_t is insufficient
56             for (int cc=LR_CELLS; cc<LR_CELLS;
57                  cc++){ // c-2 to c+2
58                 for (int t=0; t<TIMEBINS; t++){
59                     cl.qtot += buf[t][cc];
60                 }
61             }
62             // mu_x, sigma_x
63             cl.mu_p = 0;
64             cl.mu_t = 0;
65             cl.sigma_p = 0;
66             cl.sigma_t = 0;
67             // buf[0][pads], where c is center
68             // pad of bus region
69             for (int i=0; i<5; i++){
70                 int cc = c-2;
71                 cl.mu_p += ( buf[0][cc-2] +
72                           buf[0][cc-1] +
73                           buf[0][cc] +
74                           buf[0][cc+1] +
75                           buf[0][cc+2] ) * 2;
76                 cl.mu_t += ( buf[0][cc-2] +
77                           buf[0][cc-1] +
78                           buf[0][cc] +
79                           buf[0][cc+1] +
80                           buf[0][cc+2] );
81                 cl.sigma_p += ( buf[0][c-2] + 4 *
82                               buf[0][c-1] + 1 *
83                               buf[0][c+1] + 4 *
84                               buf[0][c+2] );
85                 cl.sigma_t += ( buf[0][c-2] + 4 *
86                               buf[0][c-1] + 1 *
87                               buf[0][c+1] + 4 *
88                               buf[0][c+2] );
89             }
90             // output
91             out.write(tmp);
92         }
93     }
94 }
95
96 cluster_bus_t buf[3][c+1] <
97 cl.qmax; // RD
98
99 // time (tb: 87), row (8), column
100 (8), qmax (18) and qtot (16) =
101 580bit
102 // other properties: 17bit x 4 =
103 68bit total 118bit
104
105 cl.valid = 0;
106 cl.tb = tb;
107 cl.row = r;
108 cl.col = c;
109 if (peak){
110     cl.valid = 1;
111     clusters[c] = cl;
112 }
113 #ifndef __SYNTHESIS__
114 // std::ofstream
115 // testfile("cluster-file.txt");
116 if (cl.valid == 1){
117     testfile << buf[t][i] << " ";
118     testfile << "\n";
119 }
120 #endif
121 for (int c=0; c<COLS/2; c++){
122     tmp.range(CD_BITS*(c-1),
123              CD_BITS*c) =
124     ( clusters[c+2].valid == 1 ?
125       clusters[c+2].pack() :
126       clusters[c+1].pack() );
127 }
128 // output
129 out.write(tmp);
130 }
131 }
    
```

U55で10 CRU分が入った  
290MHzで動作 → 5MSPSに対して58 clock



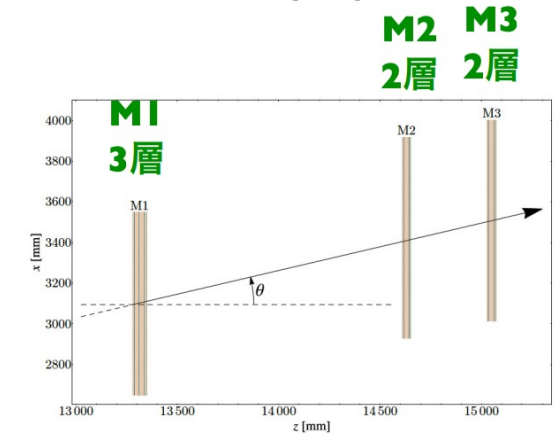
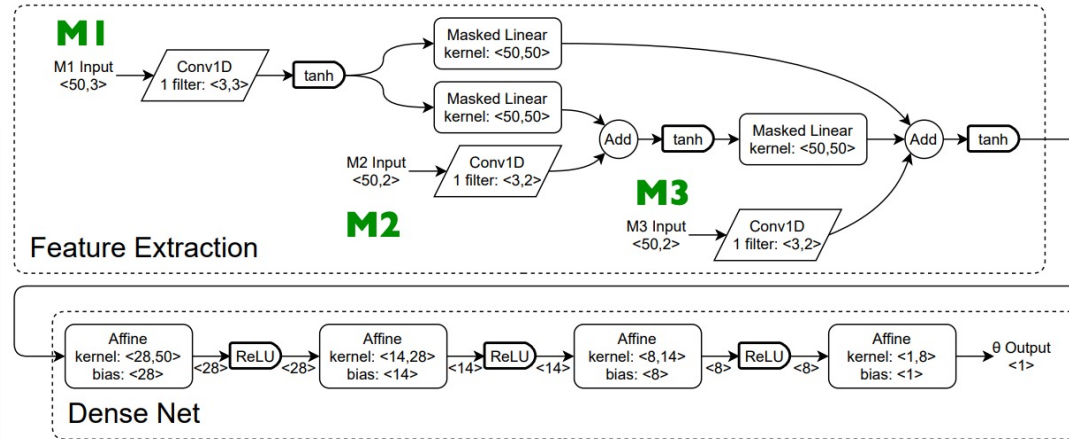
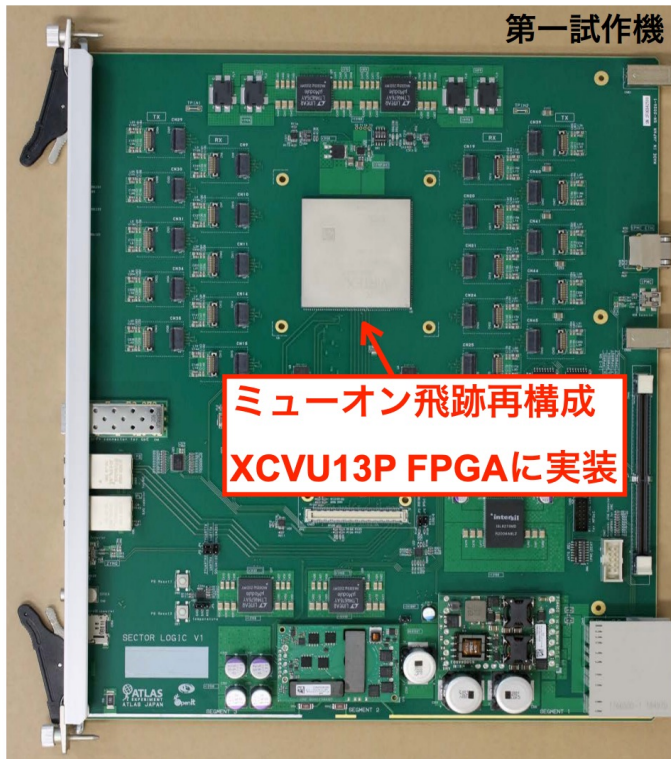
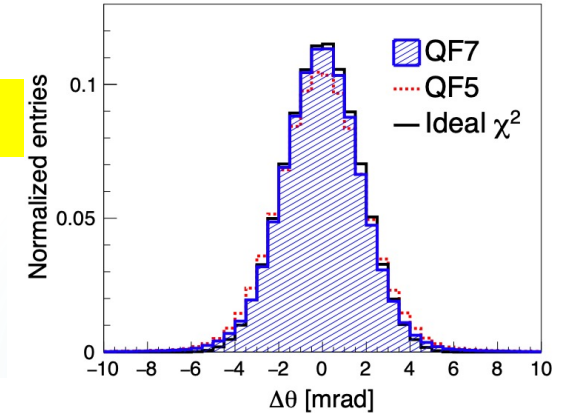
	Used	Available	%
LUT	714,307	1,303,680	54.8
FF	814,523	2,607,360	31.2
BRAM	846.5	2,016	42.0

# Online tracking using ML

## ▶ ATLAS muon tracking

Prof. Horii (Nagoya Univ.) for ATLAS

- ▶ Implementation of ML-based online tracking
- ▶ Latency < 100ns



Model	Resolution [mrad]	Latency [ns]	DSP48	LUT	FF	BRAM
BL	1.9	-	-	-	-	-
QF7	2.0	69	1,389 (45%)	34,848 (8.0%)	5,433 (0.6%)	75 (2.8%)
QF5	2.2	69	88 (2.9%)	40,039 (9.3%)	3,419 (0.4%)	75 (2.8%)
QF3	2.8	56	2 (< 0.1%)	21,682 (5.0%)	2,242 (0.3%)	75 (2.8%)






リソース使用率は、Super Logic Regionあたりの値

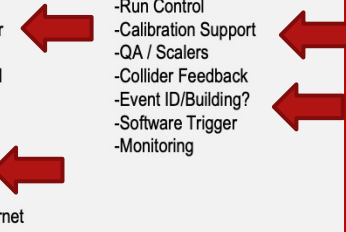
# Interest in contributing ePIC-DAQ

## ▶ Plans for ePIC

- ▶ Data processing and software trigger using hardware acceleration (FPGA, GPU, CPU)
- ▶ Vertexing and Tracking at central barrel, PID, jet reconstruction, etc...

### EPIC Electronics / DAQ Standard Component Names and Functions

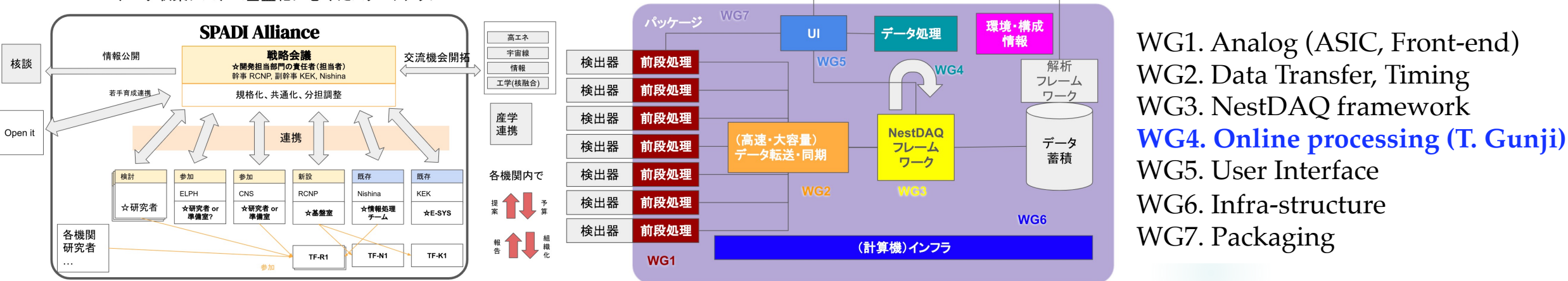
						
Name	Sensor	Adapter	Front End Board (FEB)	Readout Board (RDO)	Data Aggregation Module (DAM)	Computing
Sharing	Detector Specific	Detector Specific	Detector Specific	Few Variants	Common	Common
Function	-Multi-Channel Sensor	-HV/Bias distribution -HV divider -Interconnect routing	-Amplification -Shaping -Digitization -Zero Suppression	-Communication -Aggregation -Formatting -Data Readout -Config & Control -Clock & Timing	-Computing Interface -Aggregation -Software Trigger -Clock & Timing -Config & Control	-Data buffering and sinking -Run Control -Calibration Support -QA / Scalers -Collider Feedback -Event ID/Building? -Software Trigger -Monitoring
Attributes	-MAPS -AC-LGAD -MCP-PMT -SiPM -LAPPD	-Sensor Specific -Passive	-ASIC/ADC -Discrete -Serial Link	-FPGA -Fiber Link	-Large FPGA -PCIe -Potentially Ethernet	



# SPADI-Alliance

- ▶ “Signal Processing and Data Infrastructure – Alliance (SPADI-Alliance)” in Japan NP community to build and share common infrastructure for Streaming Readout at RIBF, J-PARC, etc...
- ▶ ASICs, FECs, streaming software, hardware acceleration
- ▶ <https://www.rcnp.osaka-u.ac.jp/~spadi/>
- ▶ ~80 members in total

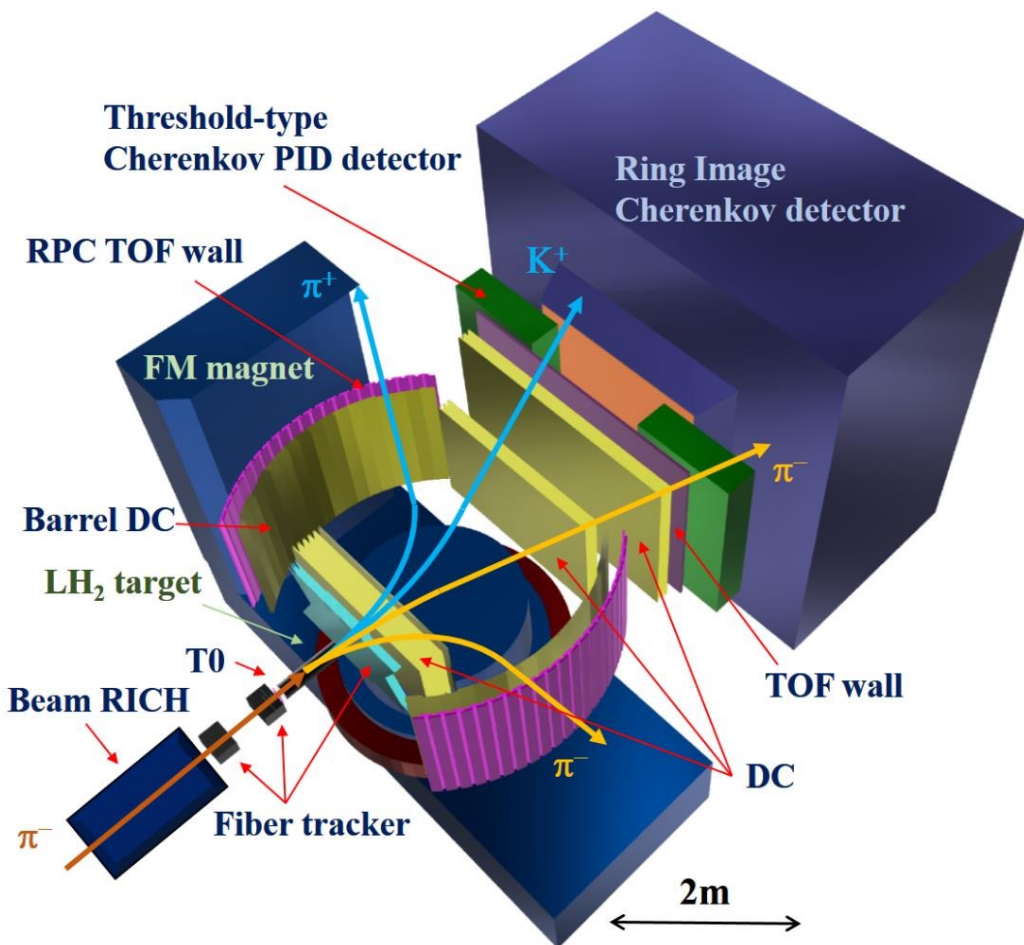
データ収集システム基盤化にむけたスタートアップ



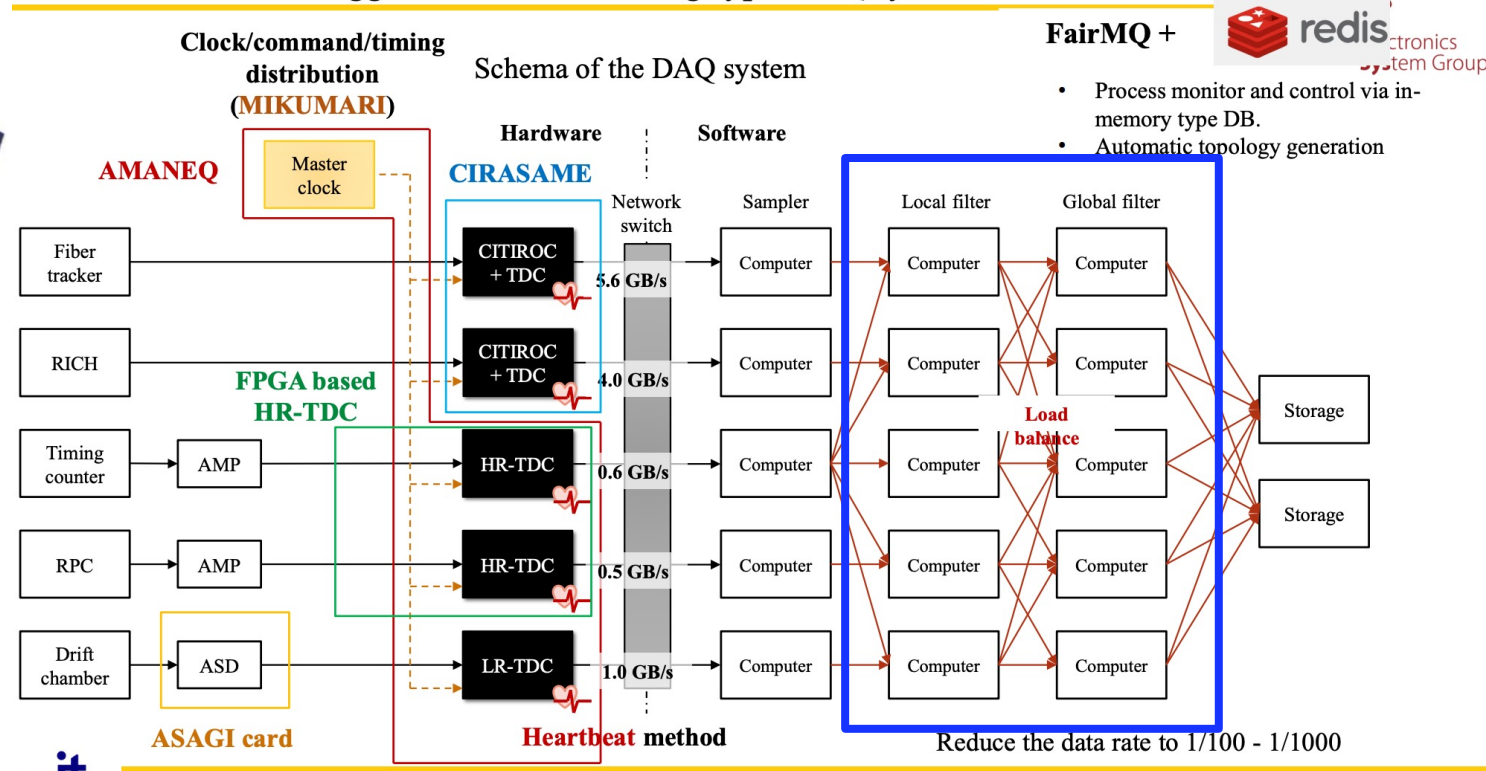
# Needs of online processing


## J-PARC E50

~10 MHz interaction rate



## Trigger-less data-streaming-type DAQ system

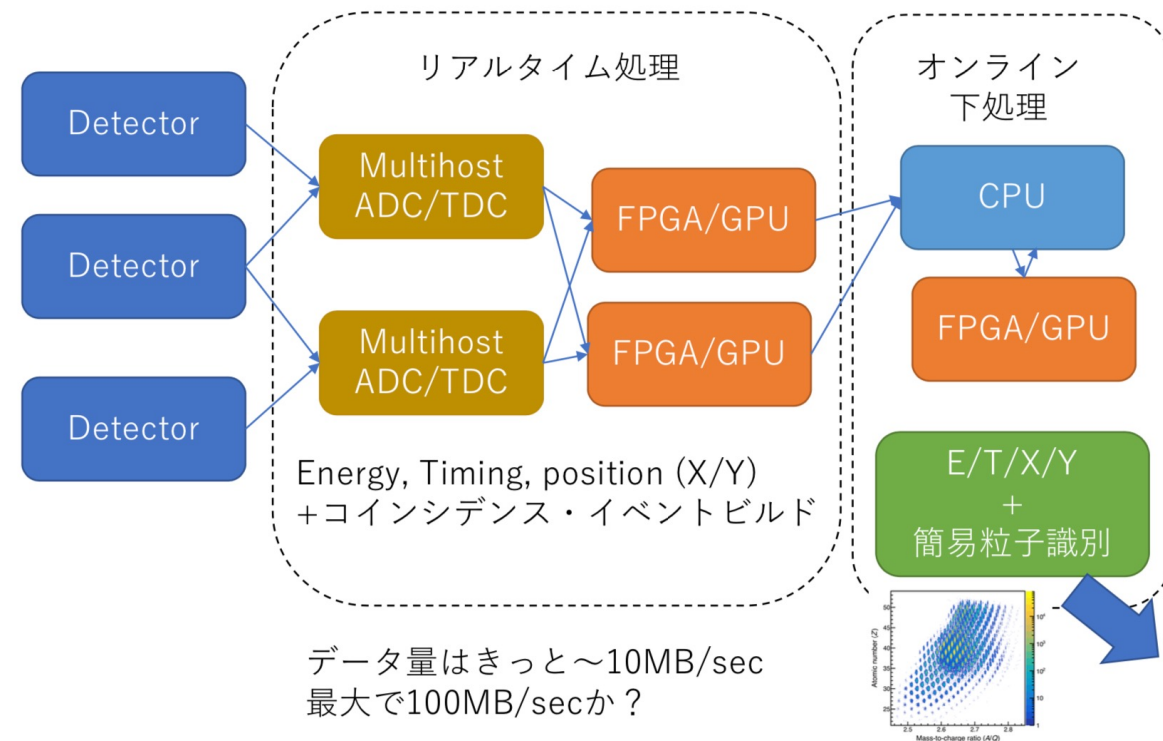
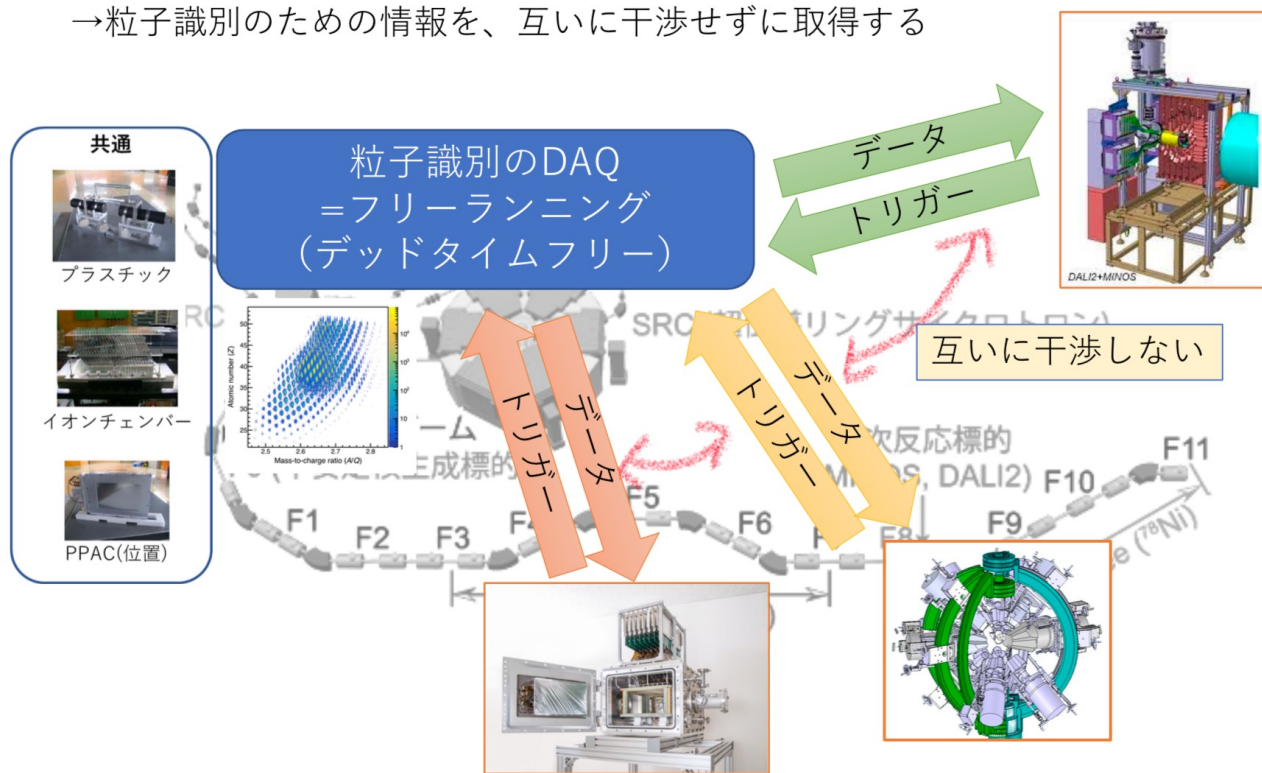


- FairMQ +  electronics system Group
- Process monitor and control via in-memory type DB.
  - Automatic topology generation

# Needs of online processing

## RIBF

今までは特定の装置が粒子識別用DAQ(CAMACやVME)を占有  
→粒子識別のための情報を、互いに干渉せずに取得する





# Interest in contributing DAQ

17

- ▶ **Plans for ePIC**
  - ▶ Data processing and software trigger using hardware acceleration (FPGA, GPU, CPU)
- ▶ **Possible work items:**
  - ▶ Run and get familiar with MC simulations
  - ▶ Benchmarking of:
    - ▶ online reconstruction (tracking, PID, etc)
    - ▶ event selections
    - ▶ usage of machine learning technique
  - ▶ Full system tests to evaluate computing models and estimate computing resources
- ▶ **Close collaboration within SPADI-A.**