DAQ prototyping: nestDAQ experience and recent development from SPADI-Alliance in Japan

TAKU GUNJI

CENTER FOR NUCLEAR STUDY, THE UNIVERSITY OF TOKYO

SPADI Alliance

Signal processing and data acquisition infrastructure alliance

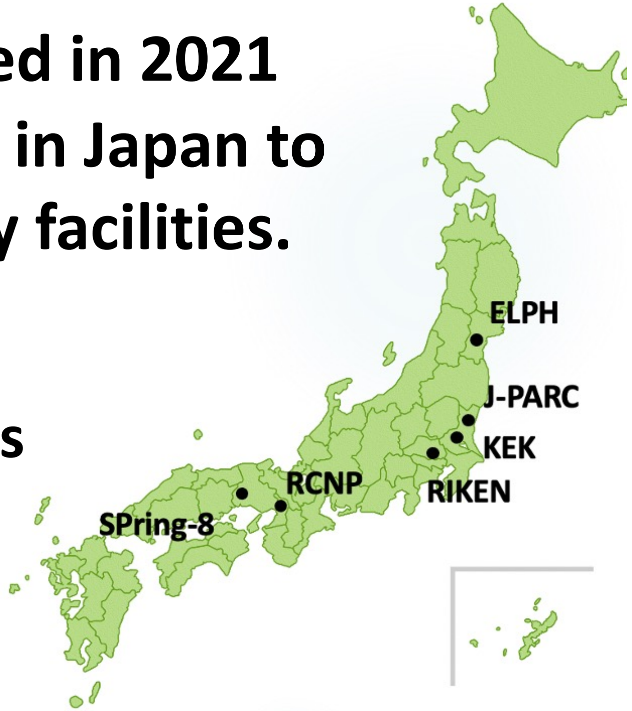
Outline

- SPADI-Alliance
- nestDAQ development
- Recent tests at J-PARC (MARQ T103 experiment)
- Next plans
- Summary

SPADI-Alliance

SPADI Alliance
Signal processing and data acquisition infrastructure alliance

- SPADI-Alliance has been formed in 2021 by nuclear physics community in Japan to realize streaming DAQ at many facilities.
  - RIBF, RCNP, J-PARC, RARIS, …
  - >120 researchers, 21 institutes



RIBF@RIKEN

RCNP, Osaka

J-PARC@JAEA/KEK

RARIS, Tohoku

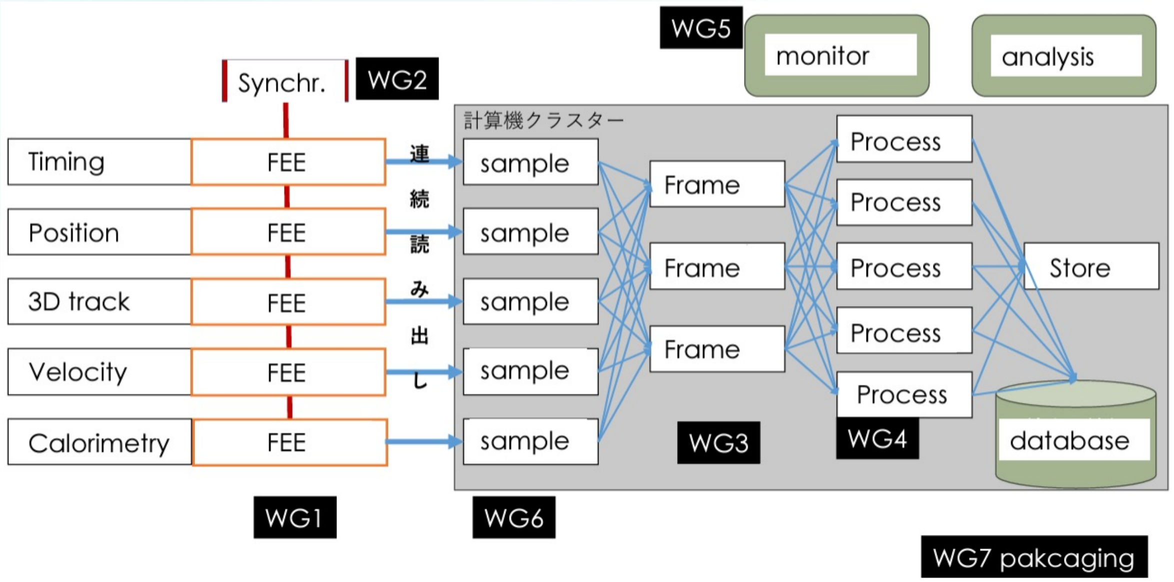
- Synergies with EIC, ALICE, and HEP experiments (Belle2 and ATLAS)
  - Joint meeting with SPADI-A/Belle2/ATLAS last week. (https://kds.kek.jp/event/51025)

SPADI-Alliance

SPADI Alliance

Signal processing and data acquisition infrastructure alliance

- 7 working groups
  - FEE, timing distribution, streaming DAQ framework,
  - Online processing (including hardware accelerators, AI/ML)
  - UI, Computing, Packaging, Offline Analysis



WG1
Frontend Electronics

Streaming type
Charge ASD board
Voltage ASD board
WF Digitizer board
Control Firmware dev.

WG2
Clock synch. / Data Transfer

General Clock Synch.
High throughput
Intra-board transfer

WG3
Acquisition software framework (NestDAQ + …)

Streaming type
FairMQ-based Scalable DAQ
Sampling, Time frame build,
Event build, Monitoring…
Format

WG4
Event processing

Acceleration using GPU/FPGA
Zero suppression
Calibration, Clustering,
Tracking, PID,

WG5
User Interface

Control, Monitor, Configure,

Trial with
SlowDash

WG6
Computing infrastr.

High throughput
Large volume
Flow and Archive
Power consumption
Interconnect
Networking
…

WG7
Packaging

Standalone system
Popularization
Standardization
Market research
User feedback

Analysis

Trial with
Artemis

WG1: ASIC and FEC

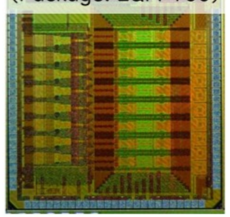
SPADI Alliance

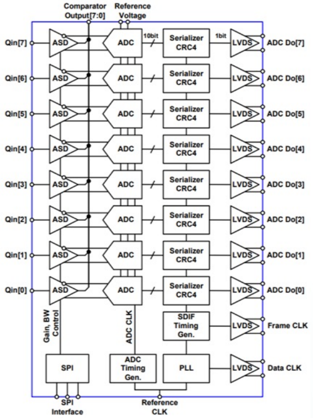
Signal processing and data acquisition infrastructure alliance

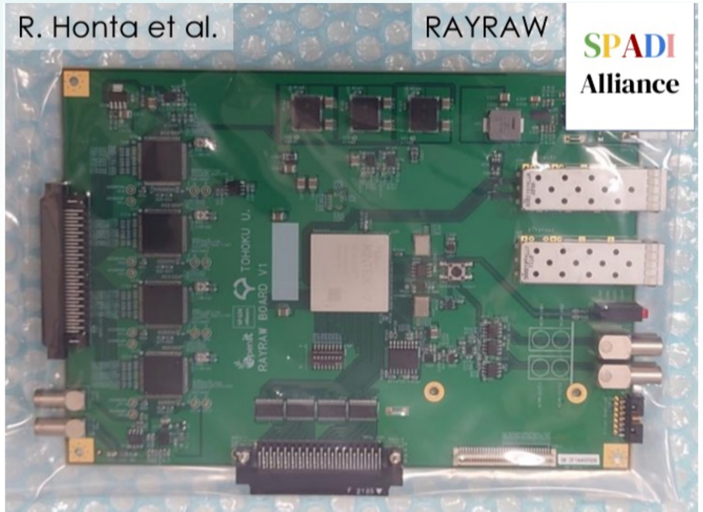
- ASIC and FEC development (supporting streaming readout)
  - Many types of FECs for MPPC, gaseous detectors, Si readout, FADC, TDC, …

MPPC ASIC Board
YAENAMI ASIC (8ch)

65 nm Si CMOS 2 mm角
(Package: LQFP100)



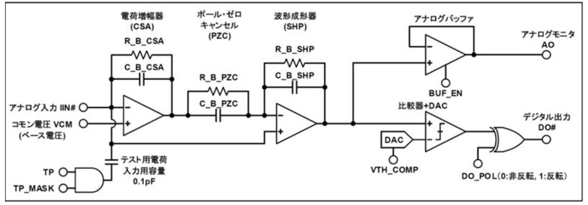


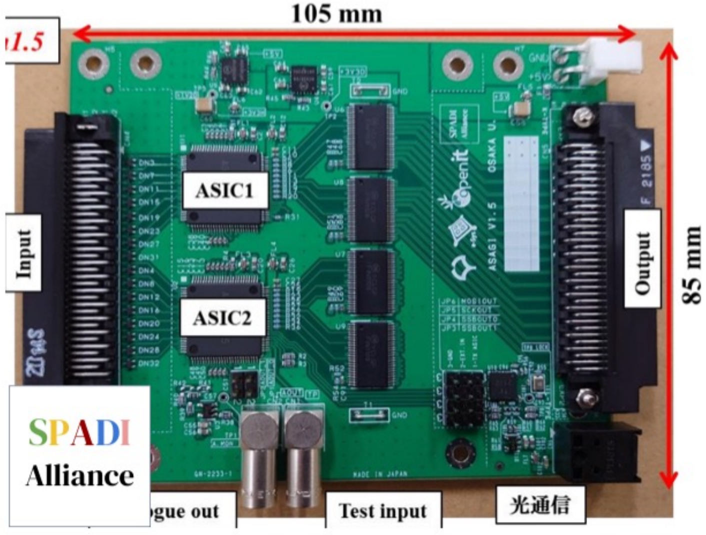

R. Honta et al.

RAYRAW

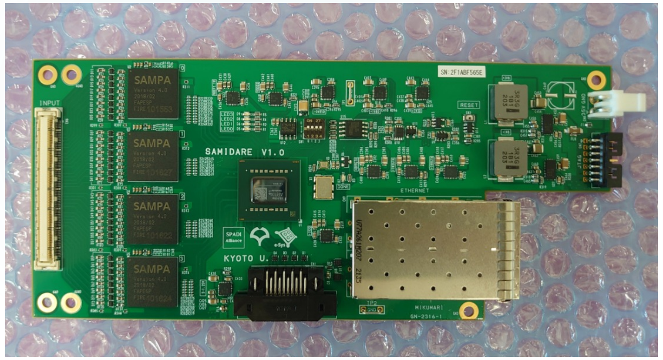
Gas chamber ASD
AGASA ASIC (16ch)

Electric circuit of 1 ASD channel





SAMPA chip board
“SAMIDARE”



High resolution
FADC MIRA

R. Honda et al.

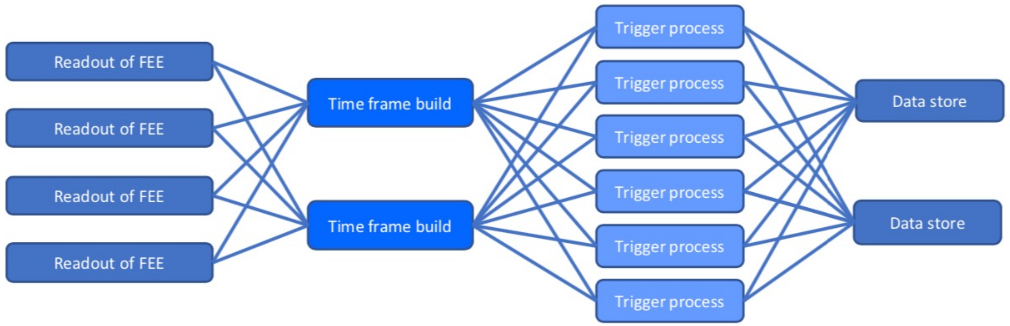
AMANEQ
HR-TDC
MIKUMARI

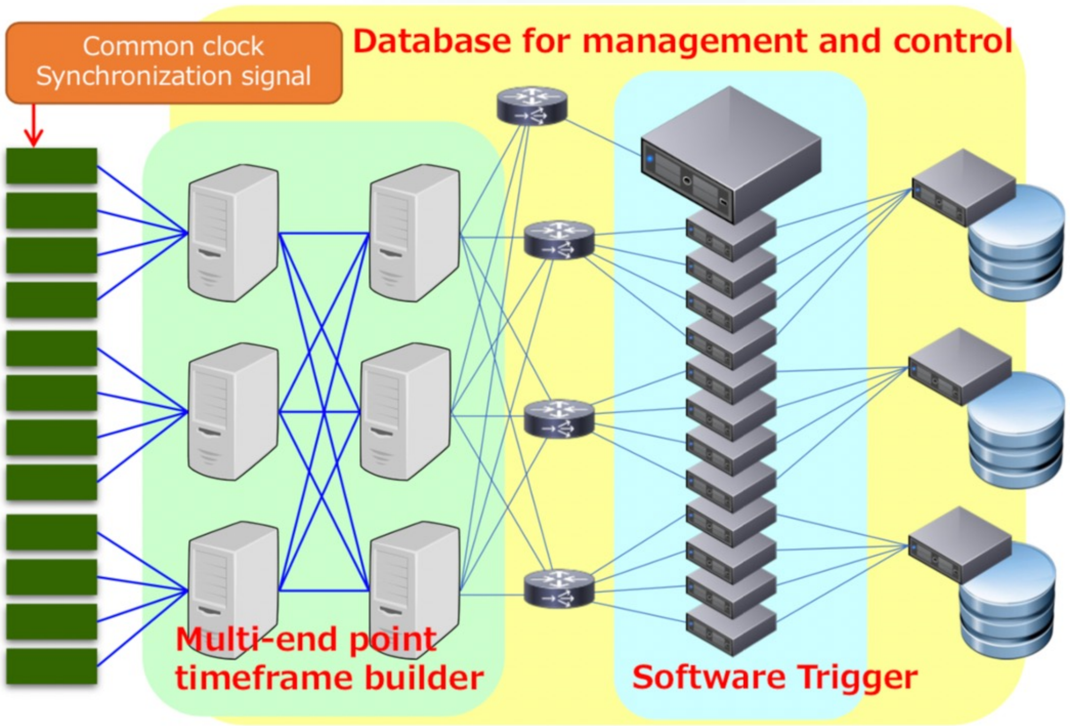
WG3: nestDAQ

- SPADI-Alliance has been developing the streaming DAQ framework.
- NestDAQ (network-based streaming DAQ)
  - overall management of dataflow and control
  - FairMQ
    - ZeroMQ as process communications
      - One to many, many to one communication
      - Message queue works as a data buffer.
  - Redis (key-value based DB)
    - Memory-oriented and fast response
    - Key-space notification
    - Pub/Sub →It can be used for control.

ØMQ

redis

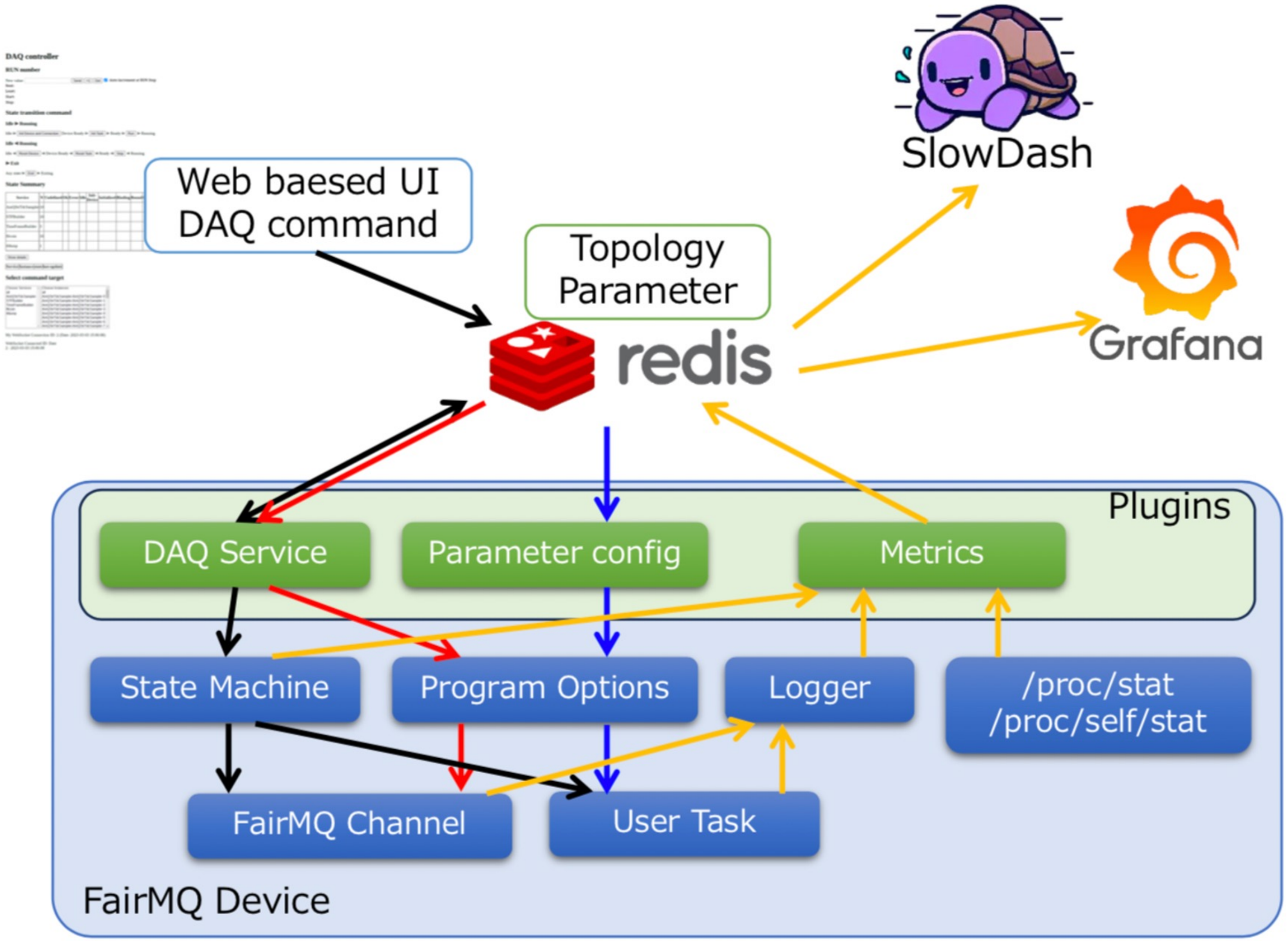




WG3: nestDAQ

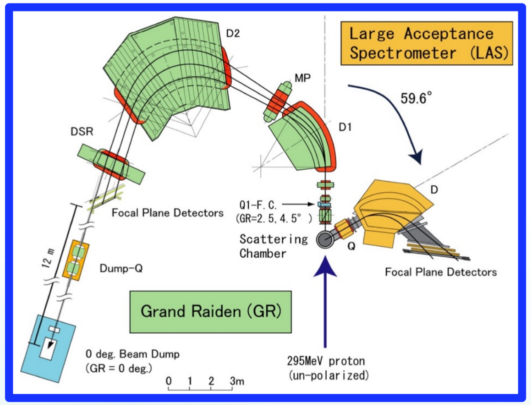
- nestDAQ process structure
- FairMQ Plugins

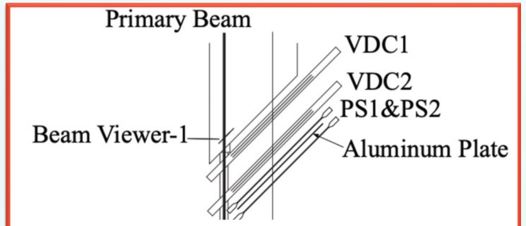
- DAQ Service Plugin
  - Run control
    - Control the state machine
    - Set the run number
  - Service discovery
    - Semi-automatic connection configuration
- Metrics Plugin
  - Grasping the processes statuses
- Parameter config Plugin
  - Read program option from the command line or the database.
  - Read device initialization parameters from the database.



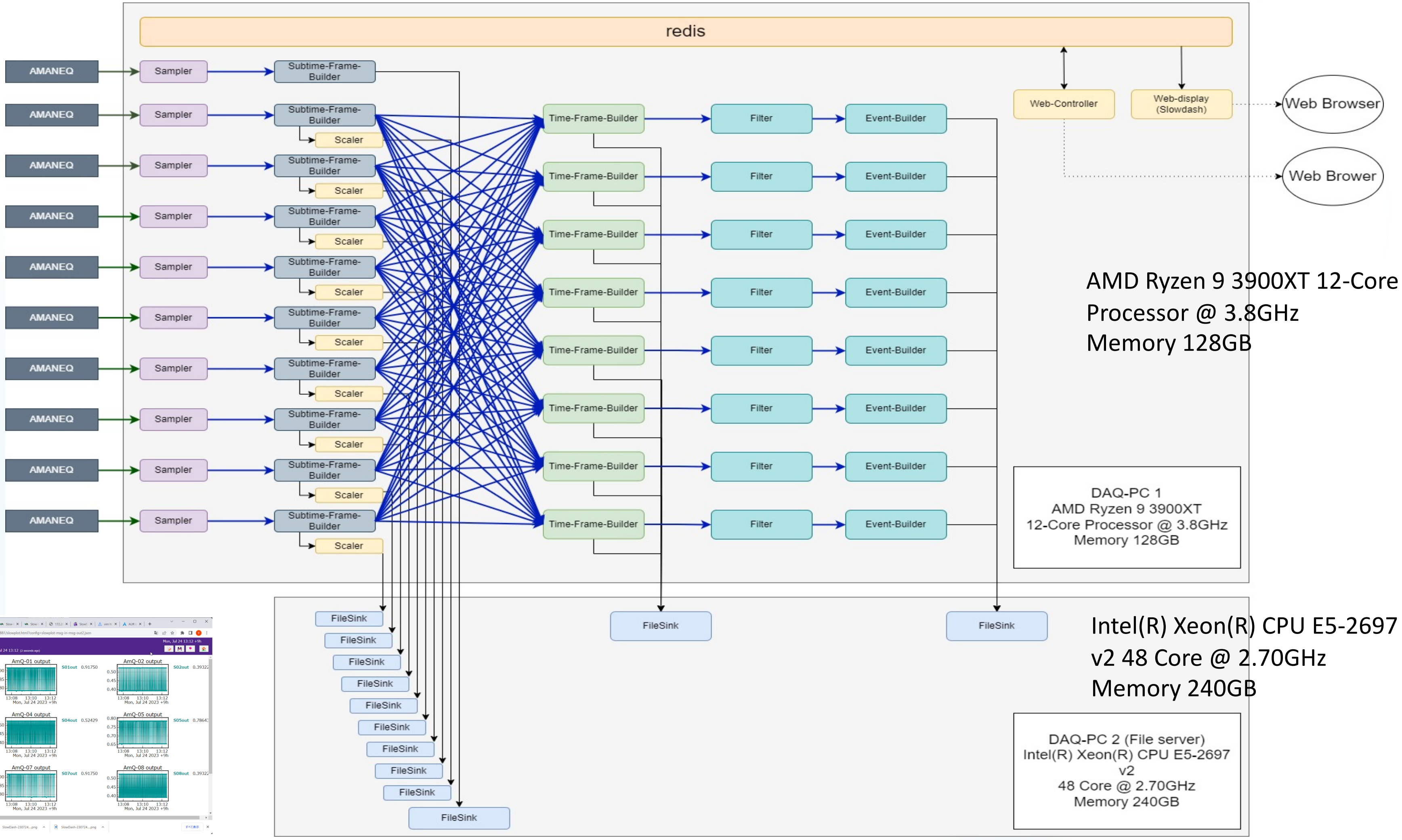
WG3: nestDAQ

▶ Example of Topology (tested 2023.7 at RCNP)





VDC (X1:192, U1:208, X2:192, U2:208)
PS (PS1: 2, PS2: 2)
RF timing, Current integrator

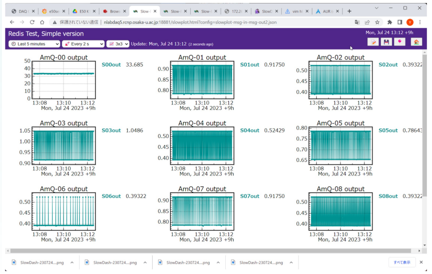


AMD Ryzen 9 3900XT 12-Core Processor @ 3.8GHz
Memory 128GB

DAQ-PC 1
AMD Ryzen 9 3900XT
12-Core Processor @ 3.8GHz
Memory 128GB

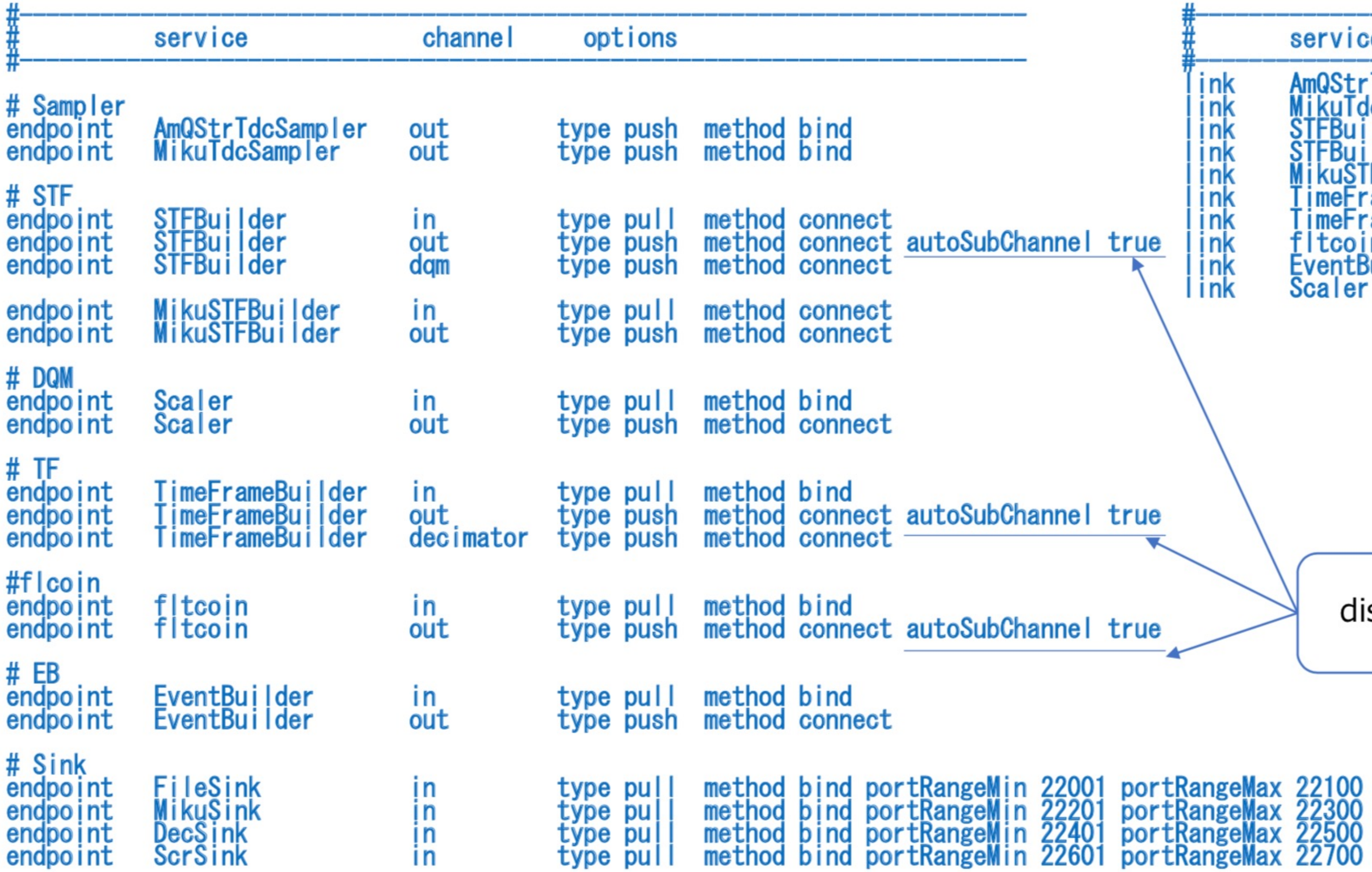
Intel(R) Xeon(R) CPU E5-2697 v2 48 Core @ 2.70GHz
Memory 240GB

DAQ-PC 2 (File server)
Intel(R) Xeon(R) CPU E5-2697 v2
48 Core @ 2.70GHz
Memory 240GB

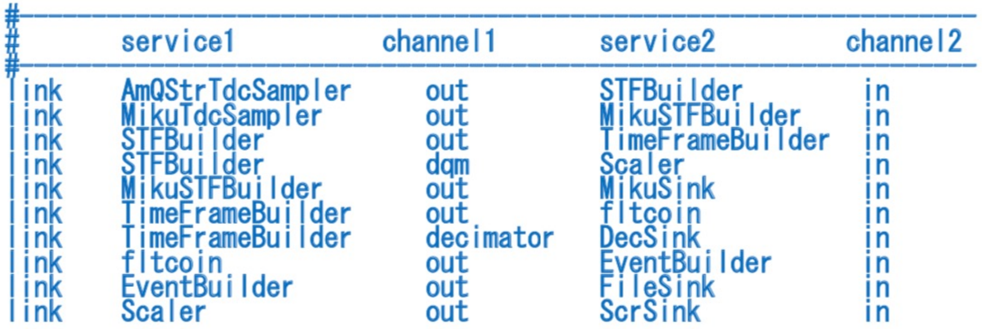


WG3: nestDAQ

▶ Corresponding topology configuration (to be stored in Redis)



```
#------------------------------------------------------------------------
#          service              channel     options
#------------------------------------------------------------------------

# Sampler
endpoint   AmQStrTdcSampler    out         type push  method bind
endpoint   MikuTdcSampler      out         type push  method bind

# STF
endpoint   STFBuilder          in          type pull  method connect
endpoint   STFBuilder          out         type push  method connect autoSubChannel true
endpoint   STFBuilder          dqm         type push  method connect

endpoint   MikuSTFBuilder      in          type pull  method connect
endpoint   MikuSTFBuilder      out         type push  method connect

# DQM
endpoint   Scaler              in          type pull  method bind
endpoint   Scaler              out         type push  method connect

# TF
endpoint   TimeFrameBuilder    in          type pull  method bind
endpoint   TimeFrameBuilder    out         type push  method connect autoSubChannel true
endpoint   TimeFrameBuilder    decimator   type push  method connect

#flcoin
endpoint   fltcoin             in          type pull  method bind
endpoint   fltcoin             out         type push  method connect autoSubChannel true

# EB
endpoint   EventBuilder        in          type pull  method bind
endpoint   EventBuilder        out         type push  method connect

# Sink
endpoint   FileSink            in          type pull  method bind portRangeMin 22001 portRangeMax 22100
endpoint   MikuSink            in          type pull  method bind portRangeMin 22201 portRangeMax 22300
endpoint   DecSink             in          type pull  method bind portRangeMin 22401 portRangeMax 22500
endpoint   ScrSink             in          type pull  method bind portRangeMin 22601 portRangeMax 22700
```

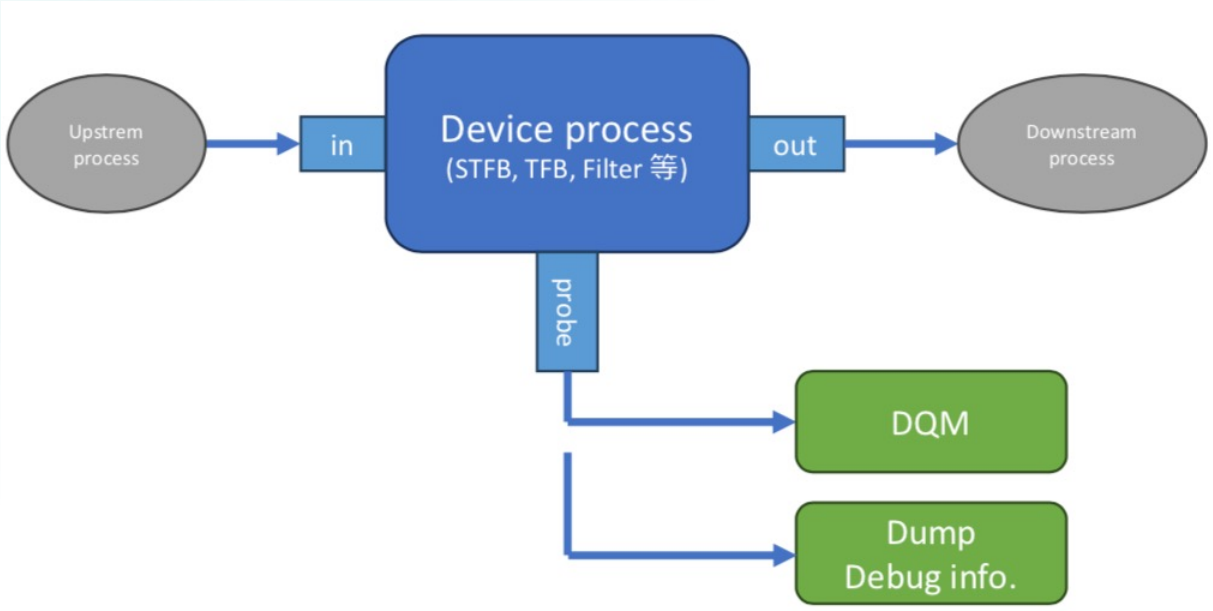


```
#--------------------------------------------------------------------
#      service1            channel1     service2           channel2
#--------------------------------------------------------------------
link   AmQStrTdcSampler    out          STFBuilder         in
link   MikuTdcSampler      out          MikuSTFBuilder     in
link   STFBuilder          out          TimeFrameBuilder   in
link   STFBuilder          dqm          Scaler             in
link   MikuSTFBuilder      out          MikuSink           in
link   TimeFrameBuilder    out          fltcoin            in
link   TimeFrameBuilder    decimator    DecSink            in
link   fltcoin             out          EventBuilder       in
link   EventBuilder        out          FileSink           in
link   Scaler              out          ScrSink            in
```

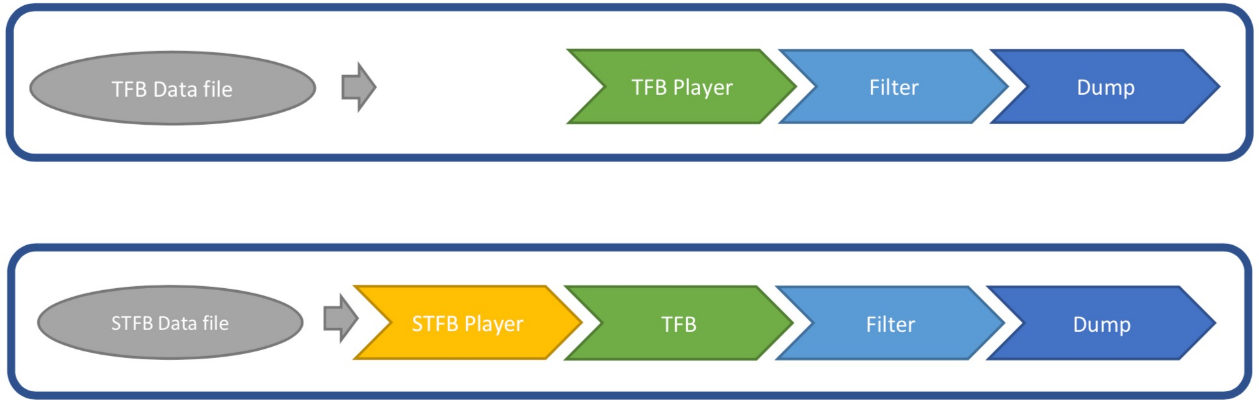
The option to use strict round-robin distribution with peer-to-peer connection for time frame building

WG3: nestDAQ

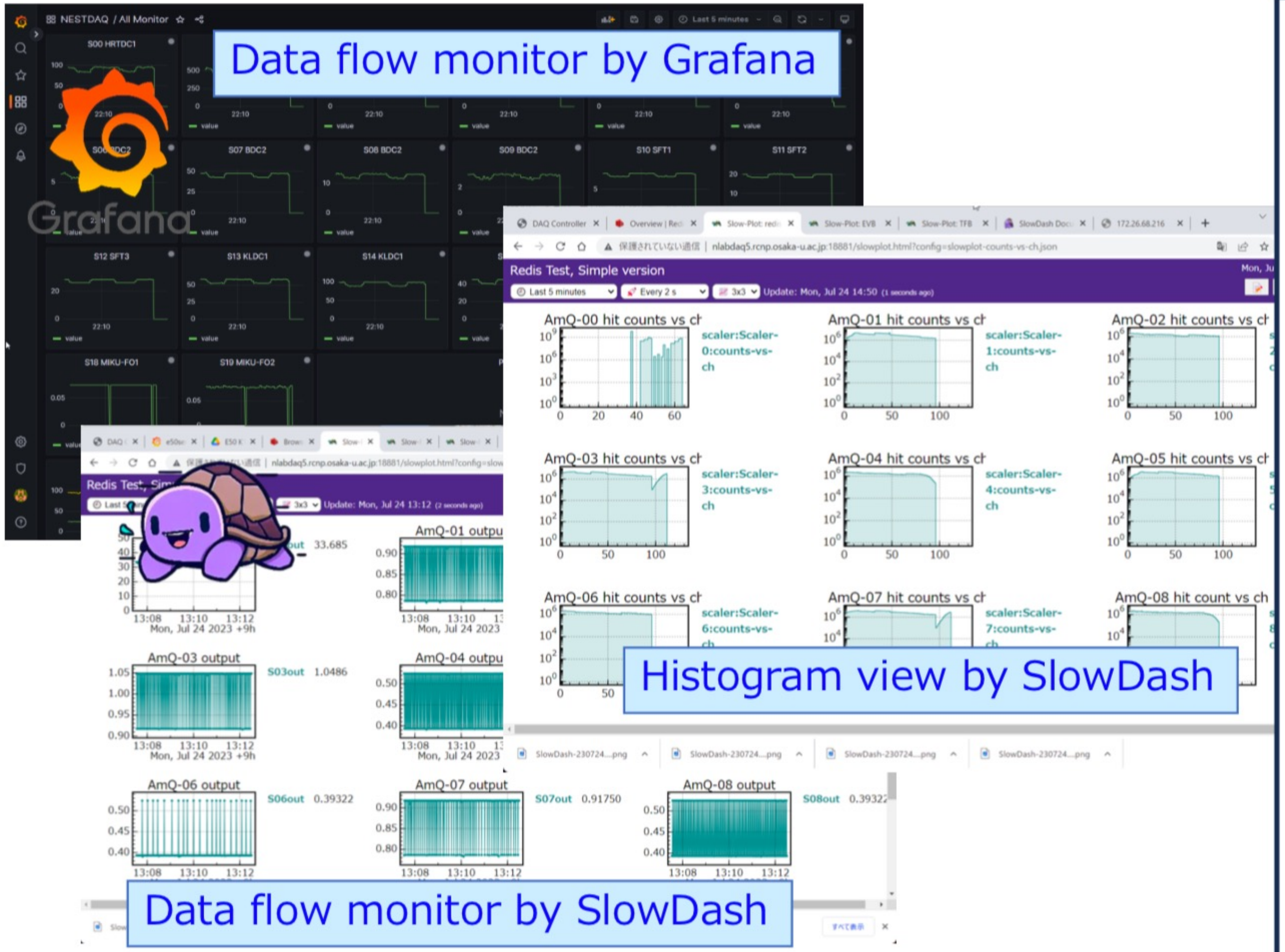
- Other supported features
  - Dataflow branch (probe port)
    - The data can be streamed separately for monitoring (ex)



- File re-player
  - process the data again in the same sequence as it was taken
  - Mandatory for developing Filter logic and benchmarking



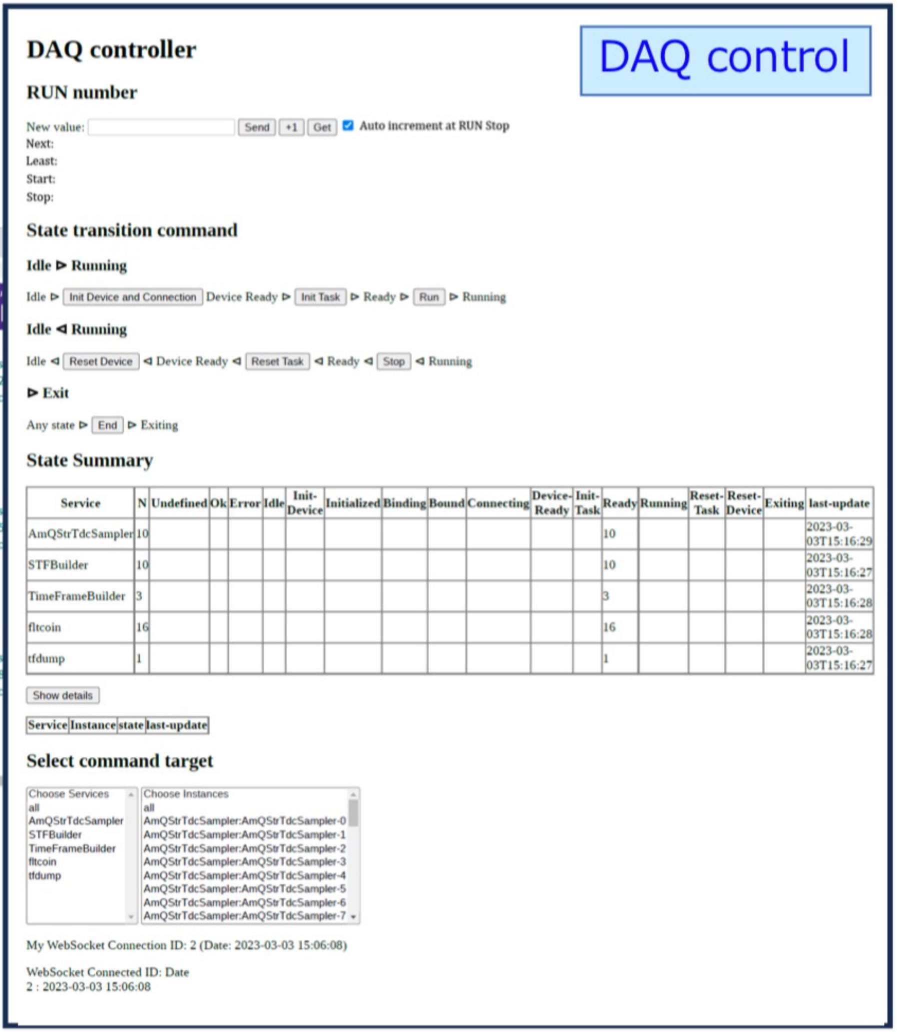
Web-UI



Data flow monitor by Grafana

Histogram view by SlowDash

Data flow monitor by SlowDash



DAQ control

DAQ controller

RUN number

New value: [   ] Send +1 Get ☑ Auto increment at RUN Stop

Next:
Least:
Start:
Stop:

State transition command

Idle ⊳ Running

Idle ⊳ Init Device and Connection Device Ready ⊳ Init Task ⊳ Ready ⊳ Run ⊳ Running

Idle ⊲ Running

Idle ⊲ Reset Device ⊲ Device Ready ⊲ Reset Task ⊲ Ready ⊲ Stop ⊲ Running

⊳ Exit

Any state ⊳ End ⊳ Exiting

State Summary

Service	N	Undefined	Ok	Error	Idle	Init-Device	Initialized	Binding	Bound	Connecting	Device-Ready	Init-Task	Ready	Running	Reset-Task	Reset-Device	Exiting	last-update
AmQStrTdcSampler	10												10					2023-03-03T15:16:29
STFBuilder	10												10					2023-03-03T15:16:27
TimeFrameBuilder	3												3					2023-03-03T15:16:28
fltcoin	16												16					2023-03-03T15:16:28
tfdump	1												1					2023-03-03T15:16:27

Show details

Service	Instance	state	last-update

Select command target

Choose Services
all
AmQStrTdcSampler
STFBuilder
TimeFrameBuilder
fltcoin
tfdump

Choose Instances
all
AmQStrTdcSampler:AmQStrTdcSampler-0
AmQStrTdcSampler:AmQStrTdcSampler-1
AmQStrTdcSampler:AmQStrTdcSampler-2
AmQStrTdcSampler:AmQStrTdcSampler-3
AmQStrTdcSampler:AmQStrTdcSampler-4
AmQStrTdcSampler:AmQStrTdcSampler-5
AmQStrTdcSampler:AmQStrTdcSampler-6
AmQStrTdcSampler:AmQStrTdcSampler-7

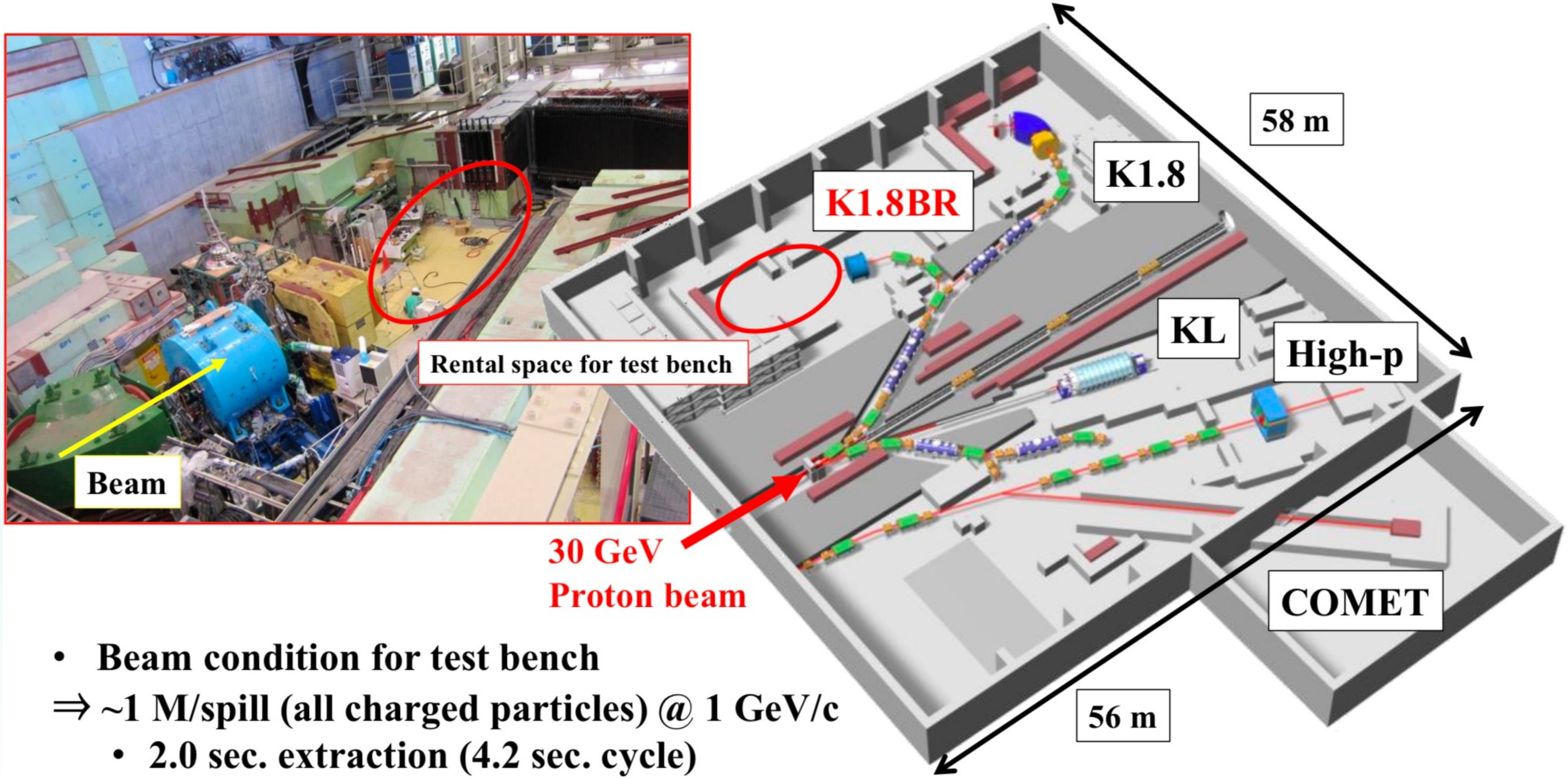
My WebSocket Connection ID: 2 (Date: 2023-03-03 15:06:08)

WebSocket Connected ID: Date
2 : 2023-03-03 15:06:08

SlowDash is a web-based visualization tool for controls, monitoring, histograms & graphs, analysis scripting, alarms…
https://github.com/slowproj/slowdash

Recent tests at J-PARC

- Testbench of streaming DAQ development at J-PARC



- Beam condition for test bench

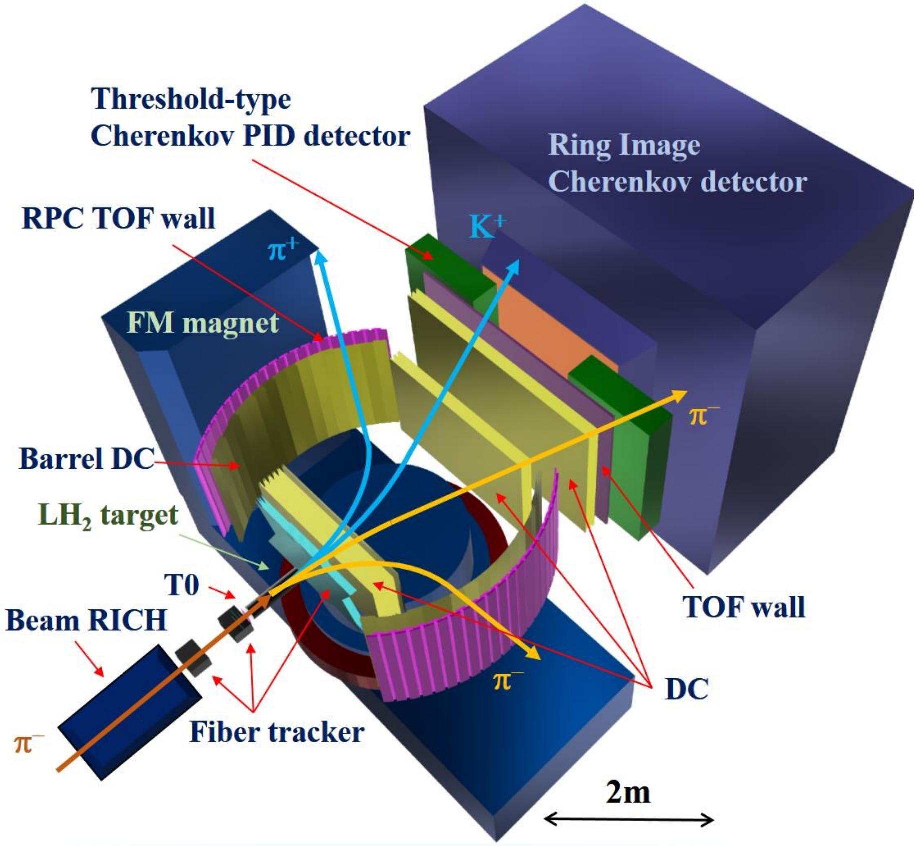
⇒ ~1 M/spill (all charged particles) @ 1 GeV/c

  - 2.0 sec. extraction (4.2 sec. cycle)

Recent tests at J-PARC

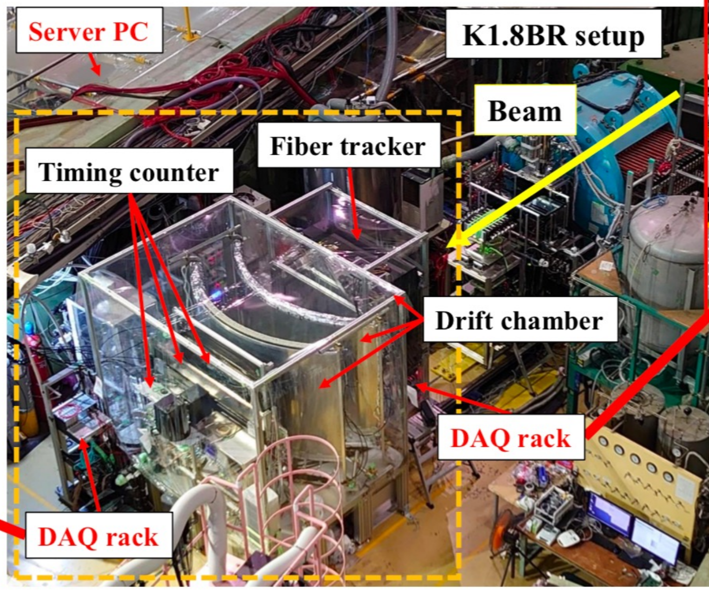
▶ Subset of E50 detectors in the Testbench (MARC T103)

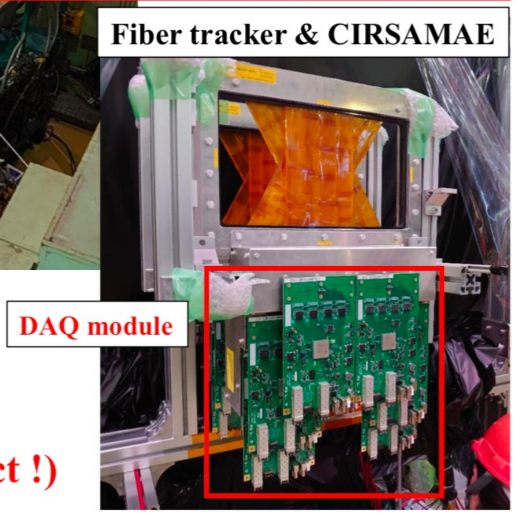
E50: Charmed baryon spectroscopy



- Missing mass method: $\pi^- + p \rightarrow \boxed{Y_c^{*+}} + D^{*-}$
  $\rightarrow K^+ \pi^- \pi^-$
  - 20 GeV/c π^- beam @ 30 Mcps

Test bench photos





- Detector readout channels: 4,142
- Front-end electronics (DAQ node)×39
- Data taking PC server×1–5

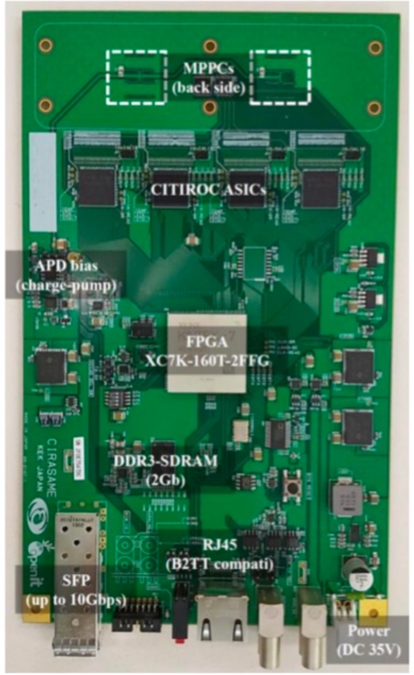
⇒ Standard nuclear physics experiment scale (Much compact !)

- **Tracking detectors**: BFT, SFT(Fiber tracker), KLDC, BDC(Drift Chamber)
- **Timing detectors**: UTOF, LTOF, DTOF, T0, (UTOF used as target), T1 from E73

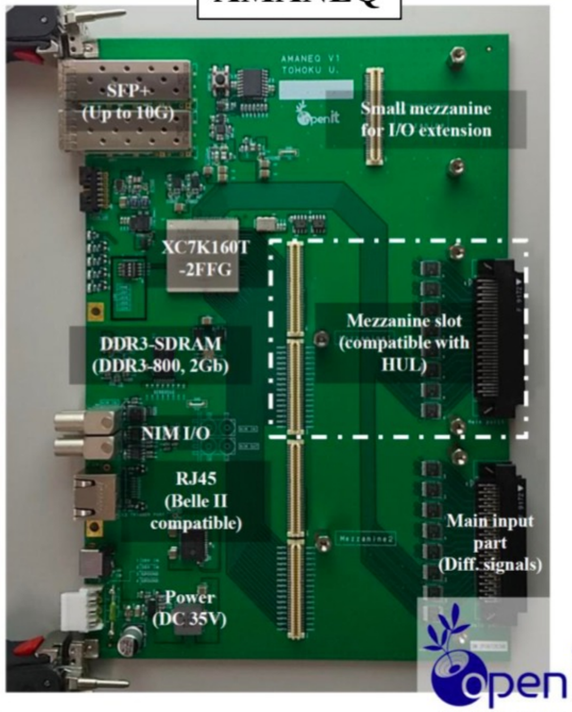
Frond-End Electronics

- Total channel ~ 4200 ch
  - MPPC (Fiber tracker) ~ 2304 ch (18 AMANEQ)
    - CIRASAME (ASIC: CITIROC) + AMANEQ (DC mezzanine)
      128 ch Low-resolution TDC (ΔT_{LSB} ~ 1 ns)
  - Timing detector (T0, TOF) ~ 128 (2 AMANEQ)
    - Amp/PMT+Discriminator + AMANEQ (HR-TDC mezzanine)
      64 ch High-resolution TDC (ΔT_{LSB} ~ 20 ps)
  - Drift chamber ~ 1920 (15 AMANEQ)
    - ASAGI(ASD) card + AMANEQ (DC mezzanine)
      ASDcard32ch→TDC128ch Low-resolution TDC (ΔT_{LSB} ~ 1 ns)
- Streaming TDC
  - Leading edge & Time-Over-Threshold(TOT)
  - Both Leading & Trailing edge: Selectable

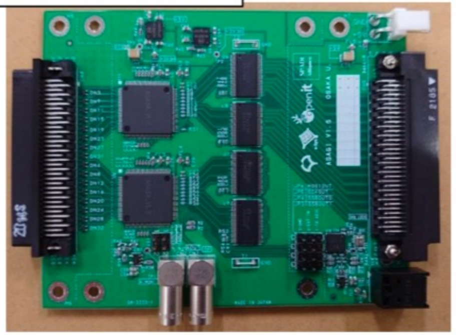
CIRASAME



AMANEQ



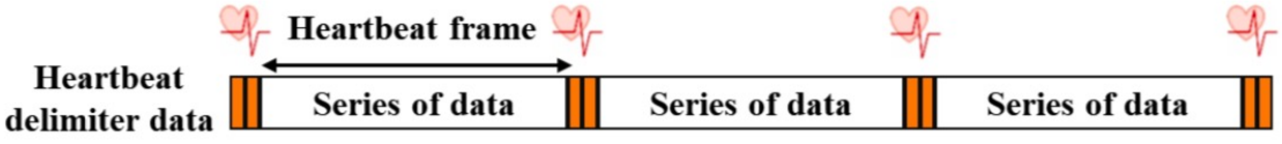
ASAGI ASD card



HR-TDC mezzanine

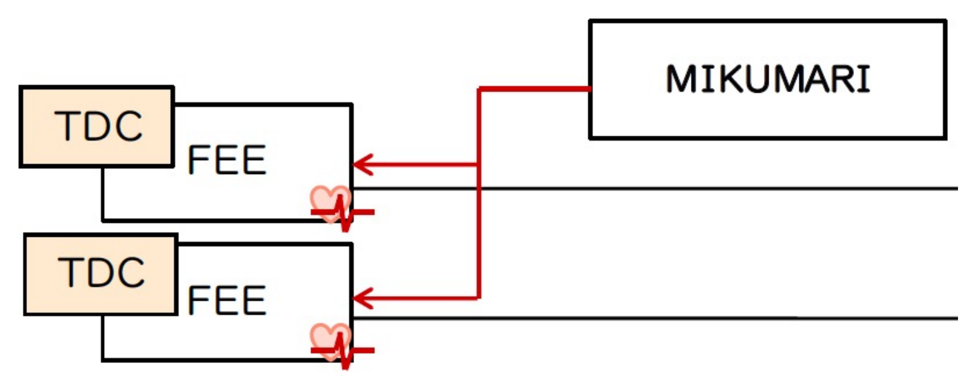
Timing distribution

- Timing synchronization: Heartbeat
  - 525 µsec period

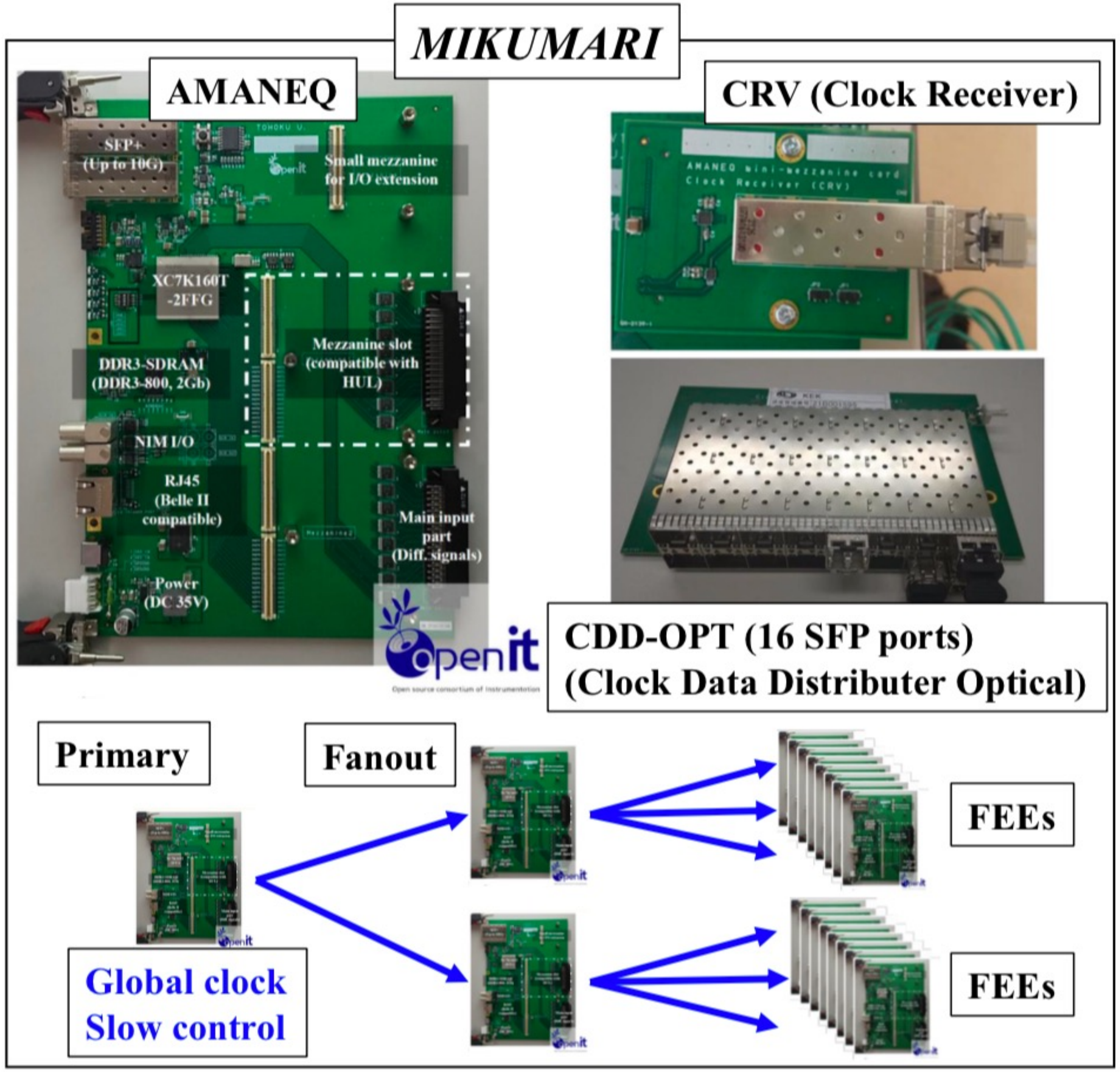


R. Honda et al., PTEP, ptab128, https://doi.org/10.1093/ptep/ptab128

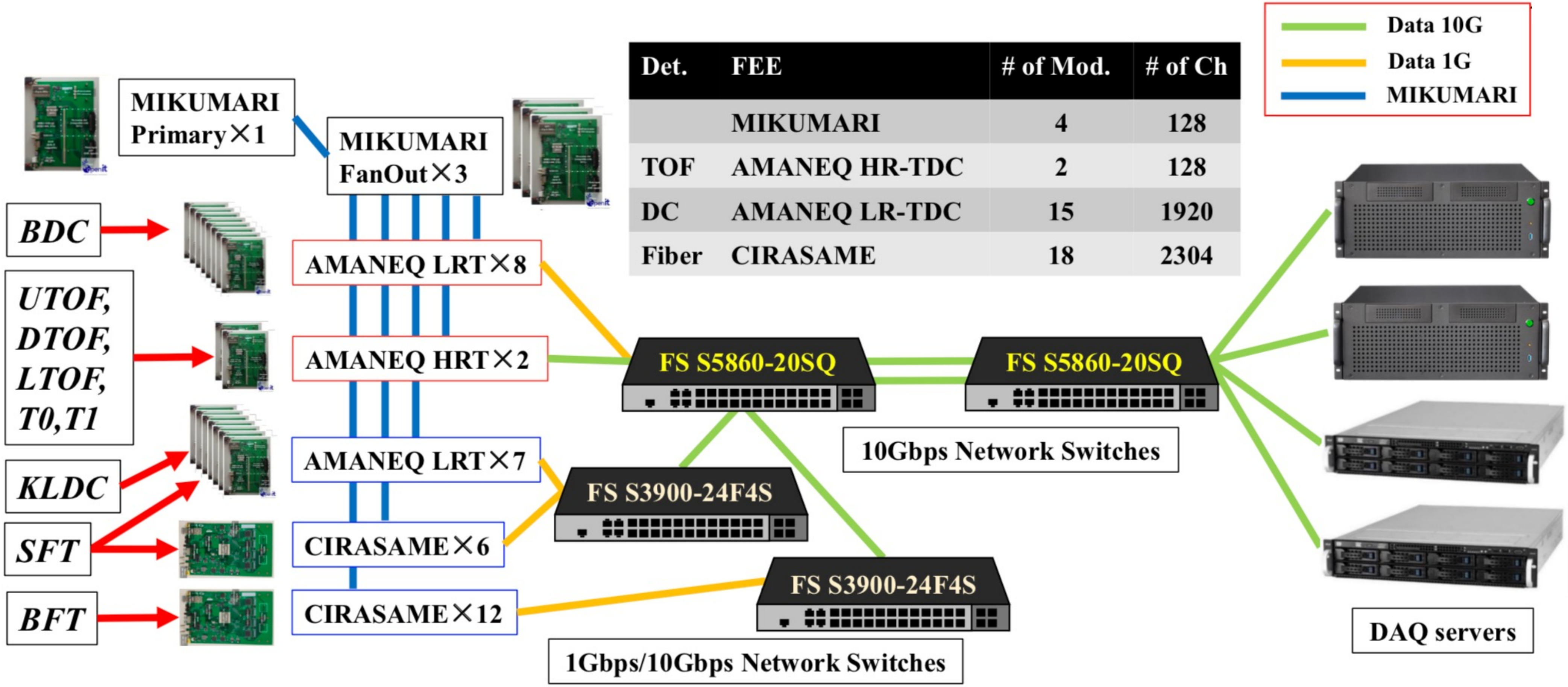
- MIKUMARI system (based on AMANEQ)



25 ps timing resolution including the synchronization precision.


MIKUMARI
AMANEQ
SFP+ (Up to 10G)
Small mezzanine for I/O extension
XC7K160T -2FFG
DDR3-SDRAM (DDR3-800, 2Gb)
Mezzanine slot (compatible with HUL)
NIM I/O
RJ45 (Belle II compatible)
Main input part (Diff. signals)
Power (DC 35V)
openit
CRV (Clock Receiver)
CDD-OPT (16 SFP ports)
(Clock Data Distributer Optical)
Primary
Fanout
FEEs
FEEs
Global clock
Slow control

DAQ nodes



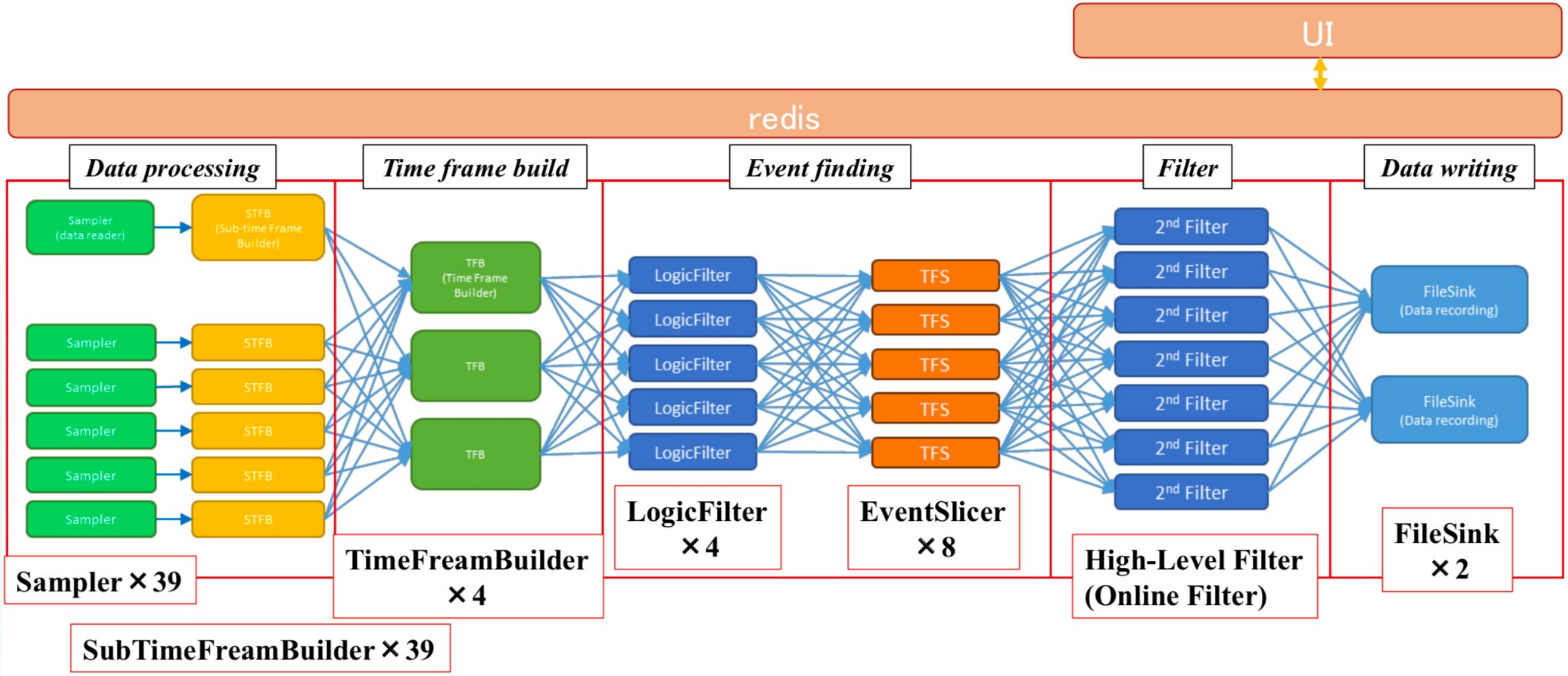
Det.	FEE	# of Mod.	# of Ch
	MIKUMARI	4	128
TOF	AMANEQ HR-TDC	2	128
DC	AMANEQ LR-TDC	15	1920
Fiber	CIRASAME	18	2304

- Upstream (SFT&BFT&KLDC): 25 nodes
- Downstream (BDC&TOFs): 10 nodes

Total 39 nodes w/ 4,142 active channels

＊AMANEQ(SrTDC×17, MIKUMARI Prim×1 + FanOut×3)×21, CIRASAME×18

nestDAQ Configuration

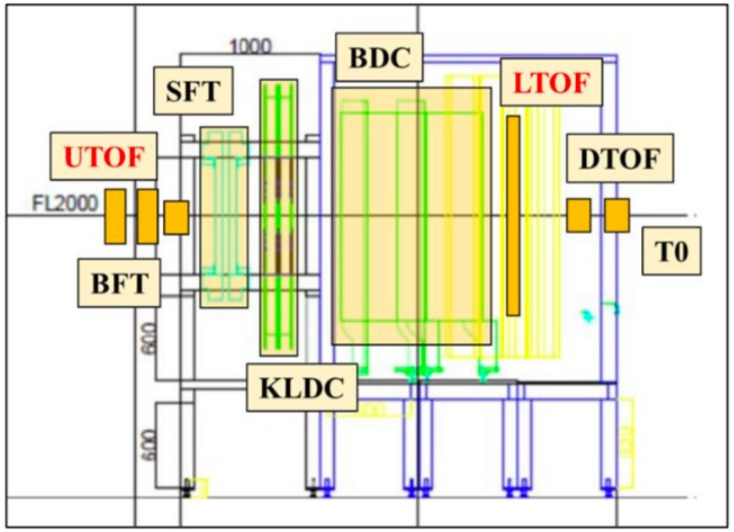


- No Filter: Sampler → STFB → TFB → FileSink
- Filtered: TFB → LogicFilter → EventSlicer → High-level Filter → FileSink

Online filters

- TFB: Reconstruction of time frame from HBF

  ＊Free streaming data taking
  - ~100% efficiency and stable operation @ ~1 M/spill
  - Averaged data rate in 4.2 sec. spill: ~130 MB/s
  - Decimation data: Pre-scaled no bias data



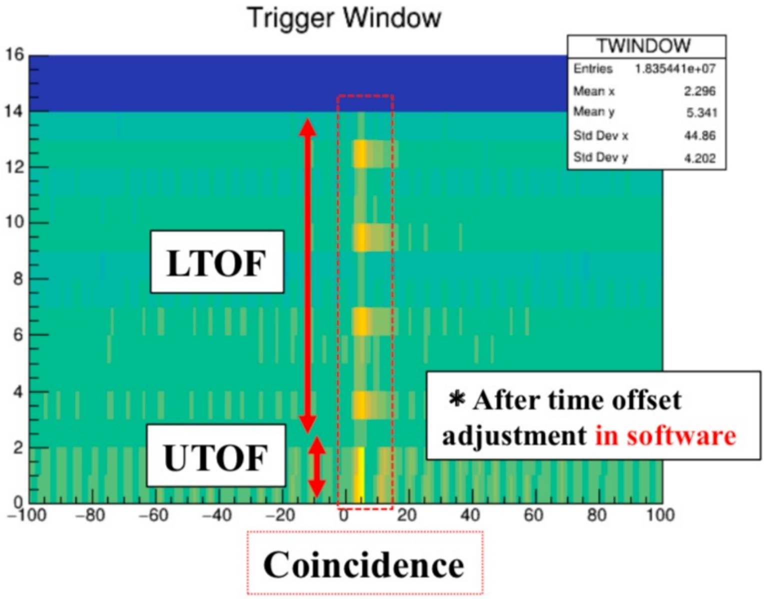
⇒ LogicFilter: Timing coincidence
- "Trigger timing" generated w/o reduction
  - UTOF × LTOF timing
  - Coincidence rate: ~200 k/spill (Reduced by detector size)

4 process
512usec x 4 = 2ms

⇒ EventSlicer: Event finding from "Trigger timing"
- Slicing window from "Trigger timing": ±1000 ns
- Timing group in Slicing window = "Event" generated

8 process
512usec x 8 = 4ms

⇒ High-level Filter: Event selection using "Event"
- "Event selection like an off-line analysis" can be performed.



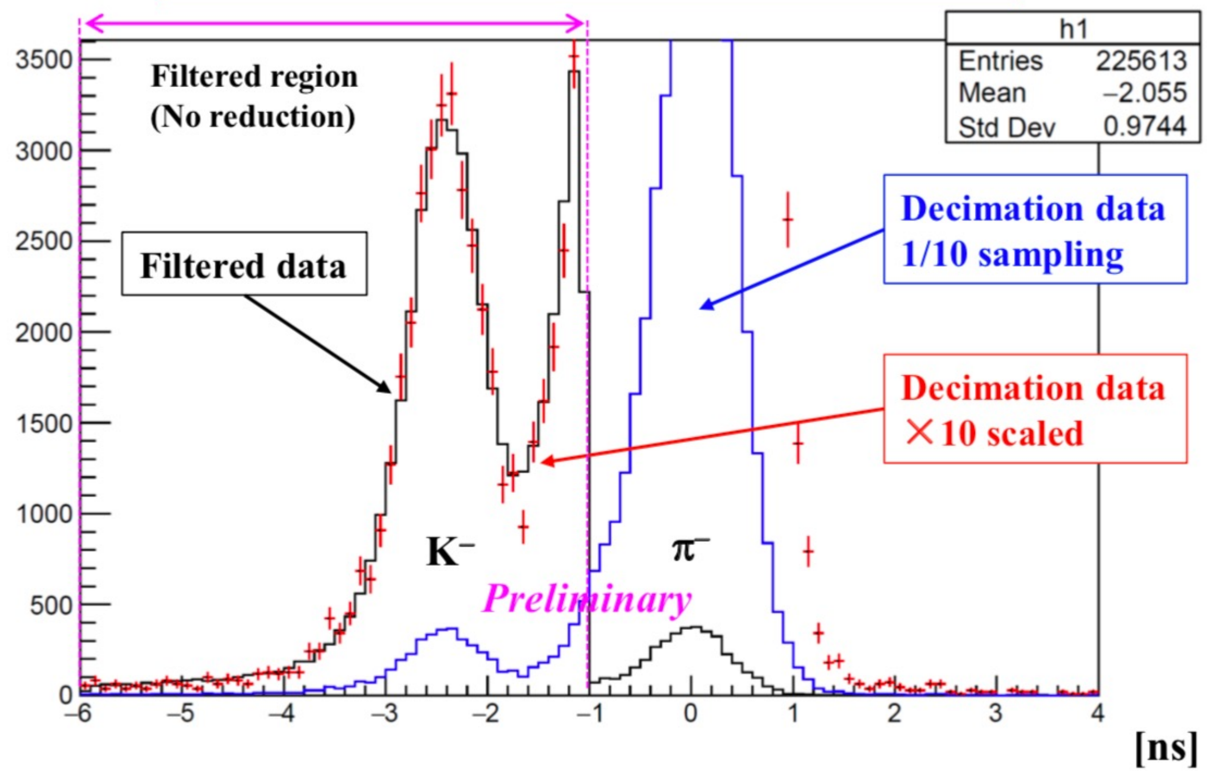
＊After time offset adjustment in software

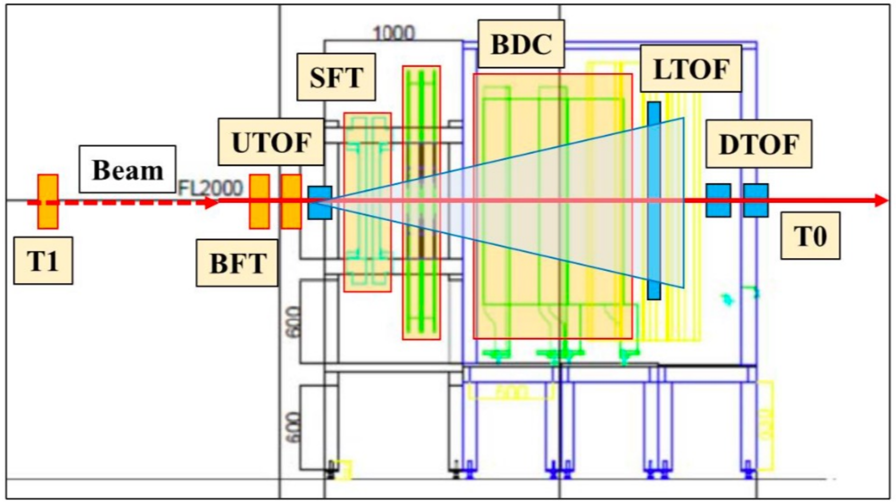
Online filters

High-level Filter: Online data selection

- Beam TOF filter: K beam selection
  - Beam TOF (5-m distance): $T_{T1:MeamTime} - T_{UTOF:MeamTime}$
- Correct Beam TOF selection ⇒ High-level filter worked well.

Beam TOF: $T_{T1:MeamTime} - T_{UTOF:MeamTime}$





- Filter conditions

done

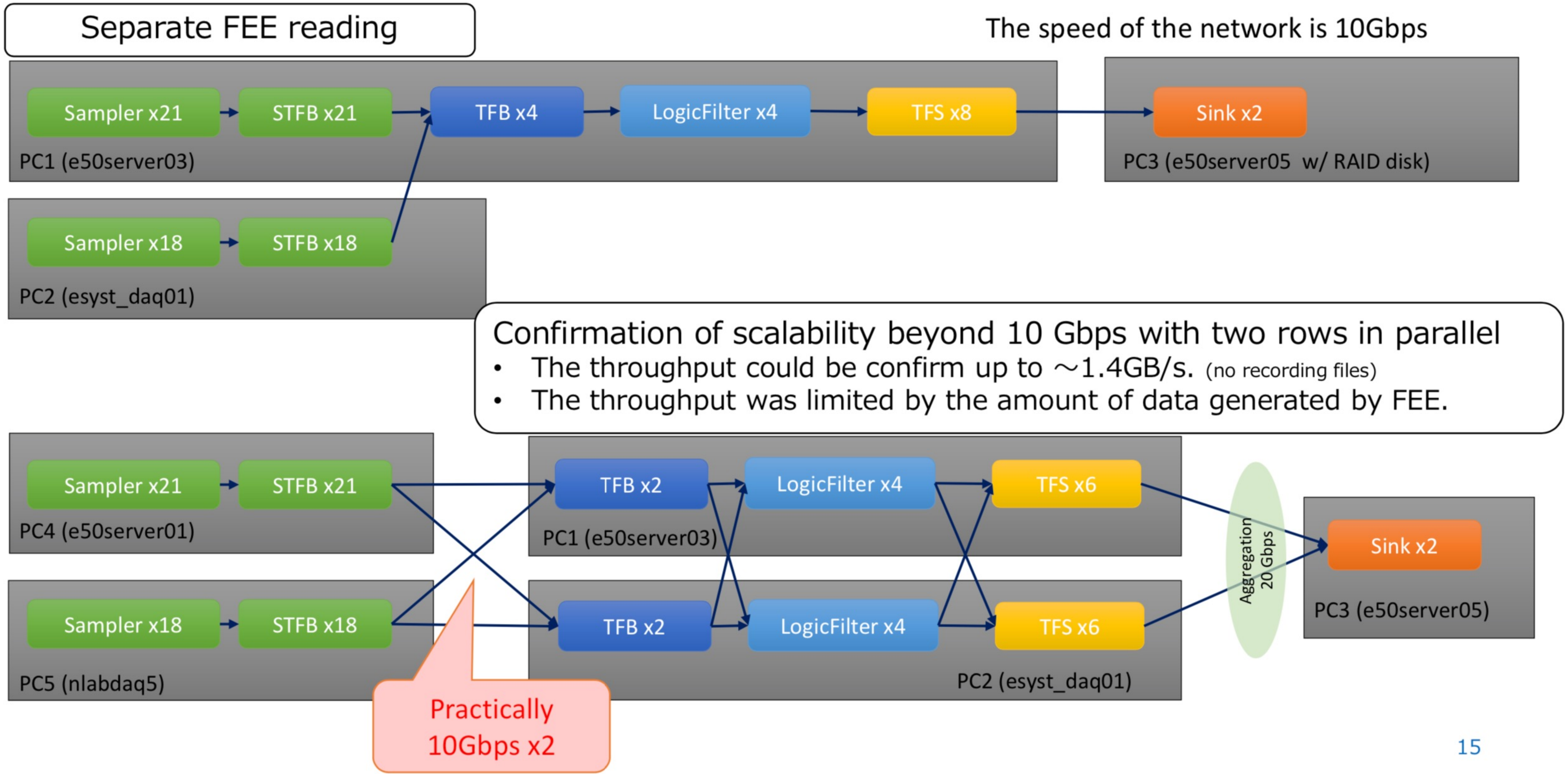
- TOF timing and time window cut
  - Timing cut with narrow timing window
  - Timing analysis with PID counter

Next
(re-play)

- DC hit pattern and correlation
  - Correlation btw DC and timing counters
- DC tracking
  - Just combination finding ⇒ Full tracking

Scalability tests

- Multi-PC setup (2:1:1, 2:2:1) and stress tests with >10Gbps



Separate FEE reading

The speed of the network is 10Gbps

Sampler x21 → STFB x21 → TFB x4 → LogicFilter x4 → TFS x8 → Sink x2

PC1 (e50server03)

PC3 (e50server05 w/ RAID disk)

Sampler x18 → STFB x18

PC2 (esyst_daq01)

Confirmation of scalability beyond 10 Gbps with two rows in parallel

- The throughput could be confirm up to ~1.4GB/s. (no recording files)
- The throughput was limited by the amount of data generated by FEE.

Sampler x21 → STFB x21

PC4 (e50server01)

Sampler x18 → STFB x18

PC5 (nlabdaq5)

TFB x2 → LogicFilter x4 → TFS x6

PC1 (e50server03)

TFB x2 → LogicFilter x4 → TFS x6

PC2 (esyst_daq01)

Aggregation 20 Gbps

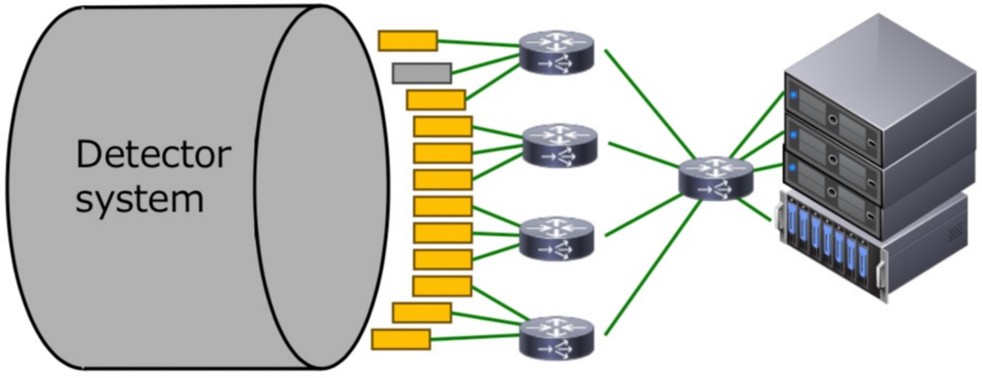
Sink x2

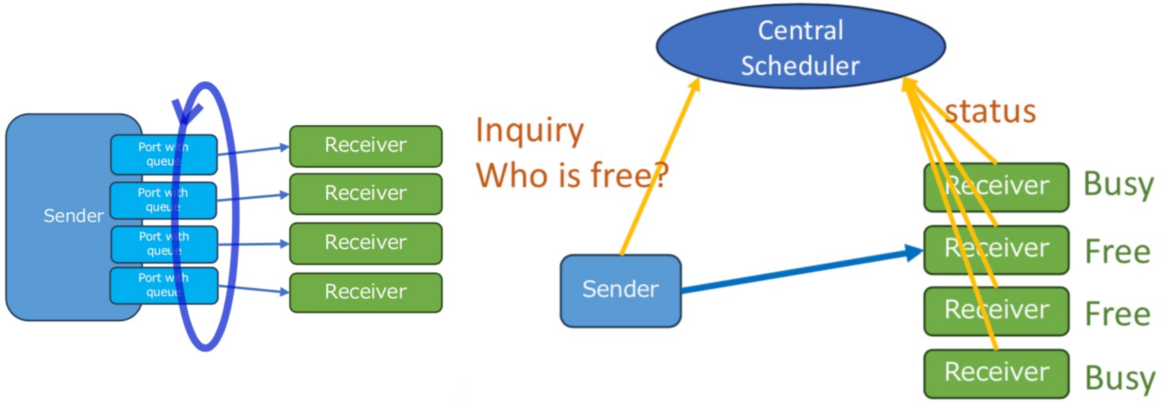
PC3 (e50server05)

Practically 10Gbps x2

Next plans for nestDAQ

- Waveform digitizer (FADC) in nestDAQ
- Log collection and display
- (Sub)Time-frame building
  - incomplete (Sub)Time-frame building
  - Dynamical plug-in/out for auto-recovery
- Adequate data discarding to reduce back-pressure
- Online filtering using accelerators (GPU, FPGA)
- Toward more large-scale system
  - Performance evaluation/validation
  - Distributed DB
  - Load balancer/scheduler
    - Round-robin or centralized system?





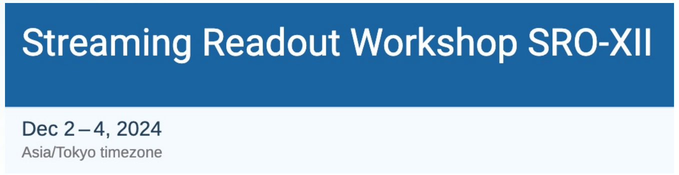
Next plans for ePIC

- Development of nestDAQ for ePIC
  - + Interface with DAM and GTM
  - + Interface with SC system and DB
  - + Interface with Calibration and EICRecon framework
- Japan is making budgetary requests to MEXT. If all goes well,
  - Increase human resources to work on streaming DAQ and computing for ePIC.
  - Purchase some computing nodes behaving as echelon0, echelon1, and echelon2, install them in BNL, and build vertical-slice setups to test full chains.
  - ~5 persons will be in BNL (or JLab) and work on developing and implementing streaming DAQ framework with BNL and JLab teams and work on testing full chains using vertical-slice setup.
  - In Japan, we will start working on echelon2.
    - Distributed online computing system (RIKEN-Tokyo-Osaka)

Summary

- We keep developing streaming DAQ software.
  - Testbench system at J-PARC
  - Implementation of nestDAQ in BDX experiment at JLab (Marco) is under discussion.
- Further developments will be planed to guarantee the scalability of the system.
- Several plans under considerations to contribute to ePIC (depending on budget…)
- Next Streaming Readout workshop will be held in Tokyo from 12.2 – 12.4.
  - Joint session with AI4EIC to discuss implementation of AI-based technologies in streaming readout.
  - We greatly welcome your coming!

https://indico.bnl.gov/event/24286/



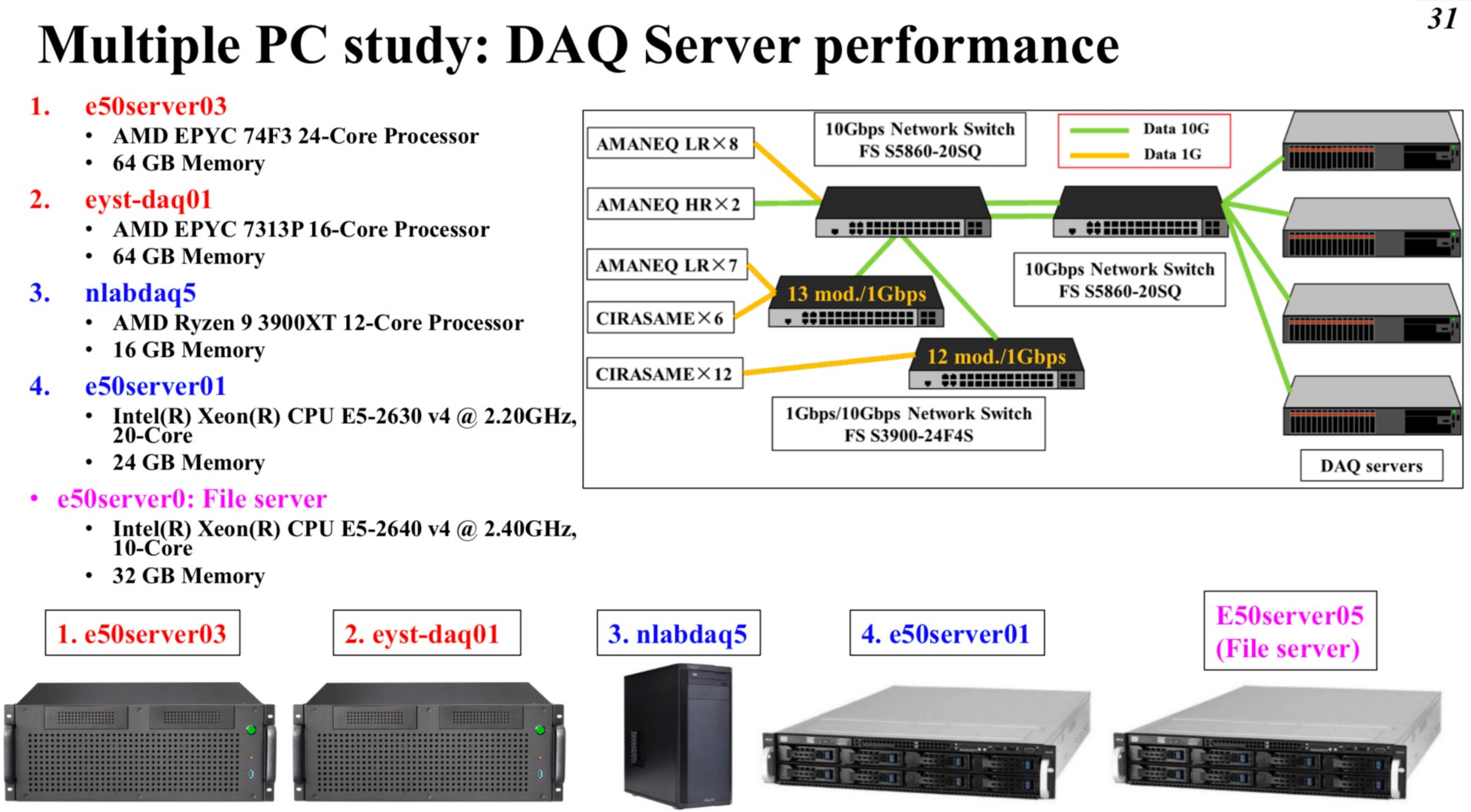
Streaming Readout Workshop SRO-XII

Dec 2 – 4, 2024

Asia/Tokyo timezone

Backup slides

Recent tests at J-PARC



31

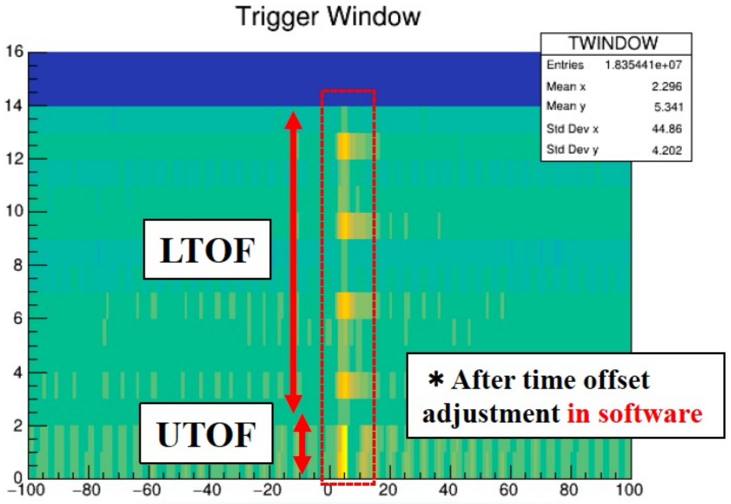
Multiple PC study: DAQ Server performance

1. e50server03
   - AMD EPYC 74F3 24-Core Processor
   - 64 GB Memory
2. eyst-daq01
   - AMD EPYC 7313P 16-Core Processor
   - 64 GB Memory
3. nlabdaq5
   - AMD Ryzen 9 3900XT 12-Core Processor
   - 16 GB Memory
4. e50server01
   - Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz, 20-Core
   - 24 GB Memory

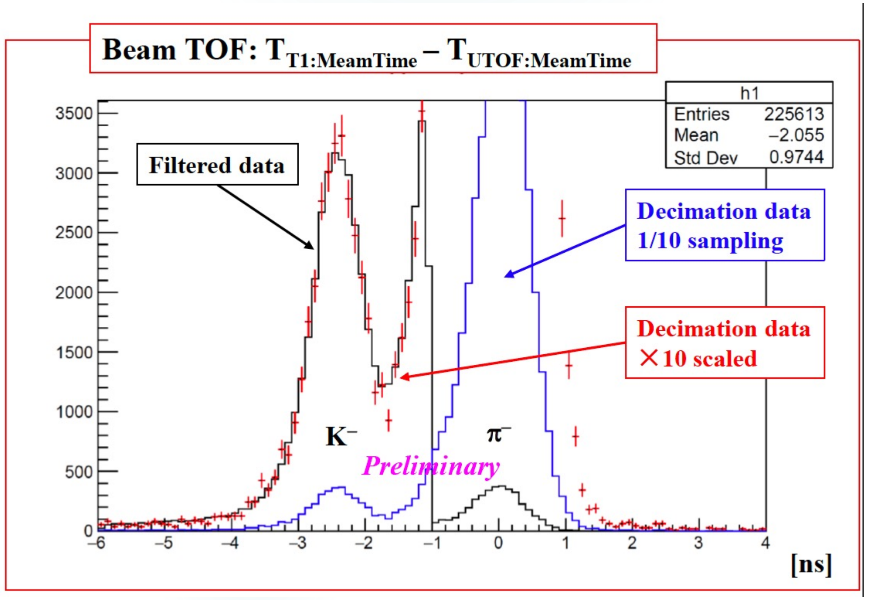
- e50server0: File server
  - Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, 10-Core
  - 32 GB Memory

Streaming DAQ and computing

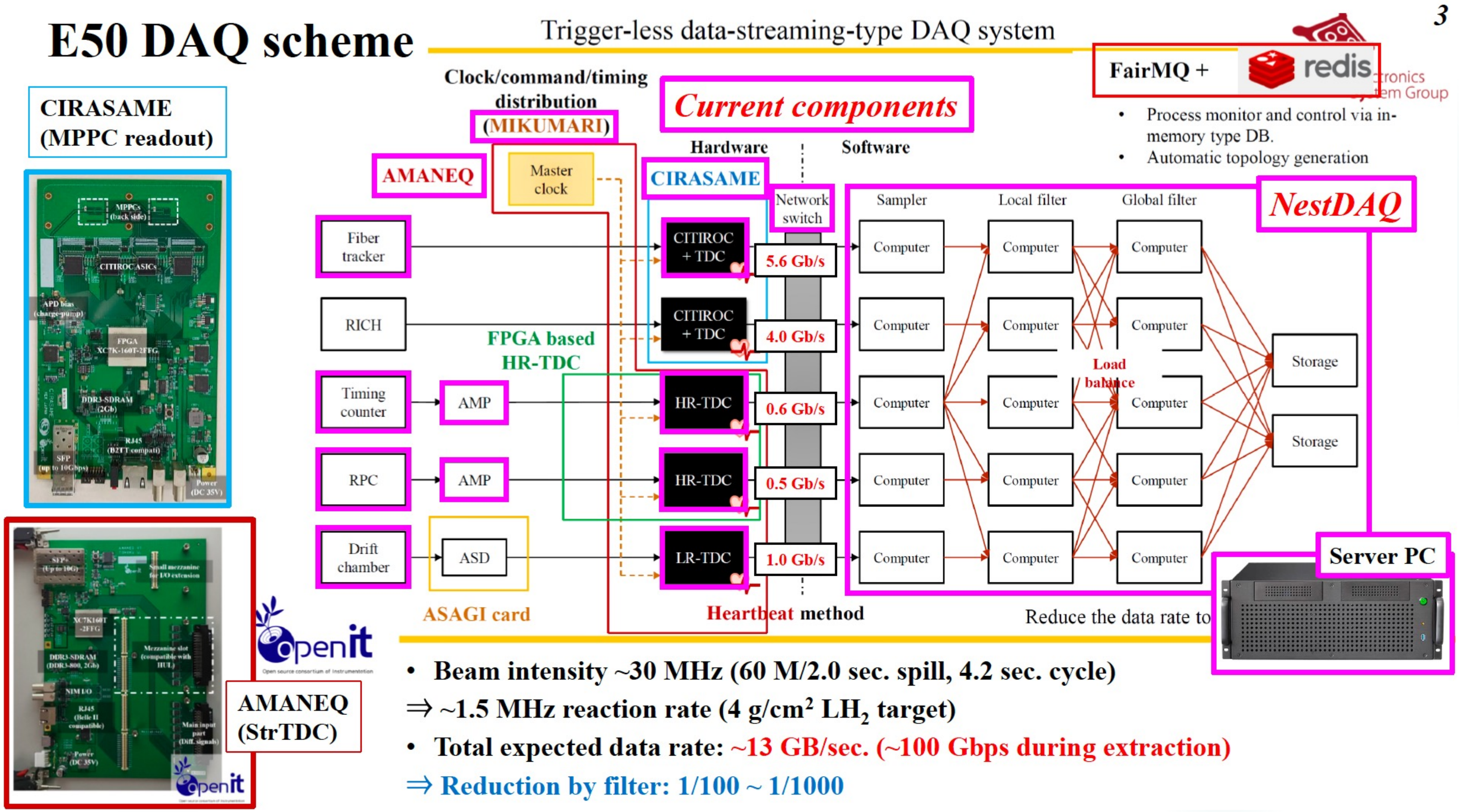
- **DAQ commissioning and studies**
  - All FEE operation: AMANEQ HR/LR-TDC, CIRASAME, MIKUMARI
  - Operation of NestDAQ with full processes
    - ~100% efficiency and stable operation @ ~1 M/spill beam flux
    - Averaged data rate during 4.2 sec. spill: 100-130 MB/s (2 HDD writing by RAID0)
  - Monitoring using SlowDashScaler readout from RBCP
  - Implementation of High-level FilterTFB→Event building→Filter→Sink
  - Data-taking PC configuration flexibility test
    - 1PC → 2PC×2PC×1PC configuration
  - High data-rate test
    - Envelopment of ~20 Gbps data rate from FEE modules
- **Discussion ongoing to deploy the nestDAQ at JLab (BDX experiment – Marco Battaglieri)**



∗ After time offset adjustment in software



Streaming DAQ and computing



E50 DAQ scheme

Trigger-less data-streaming-type DAQ system

3

FairMQ + redis

- Process monitor and control via in-memory type DB.
- Automatic topology generation

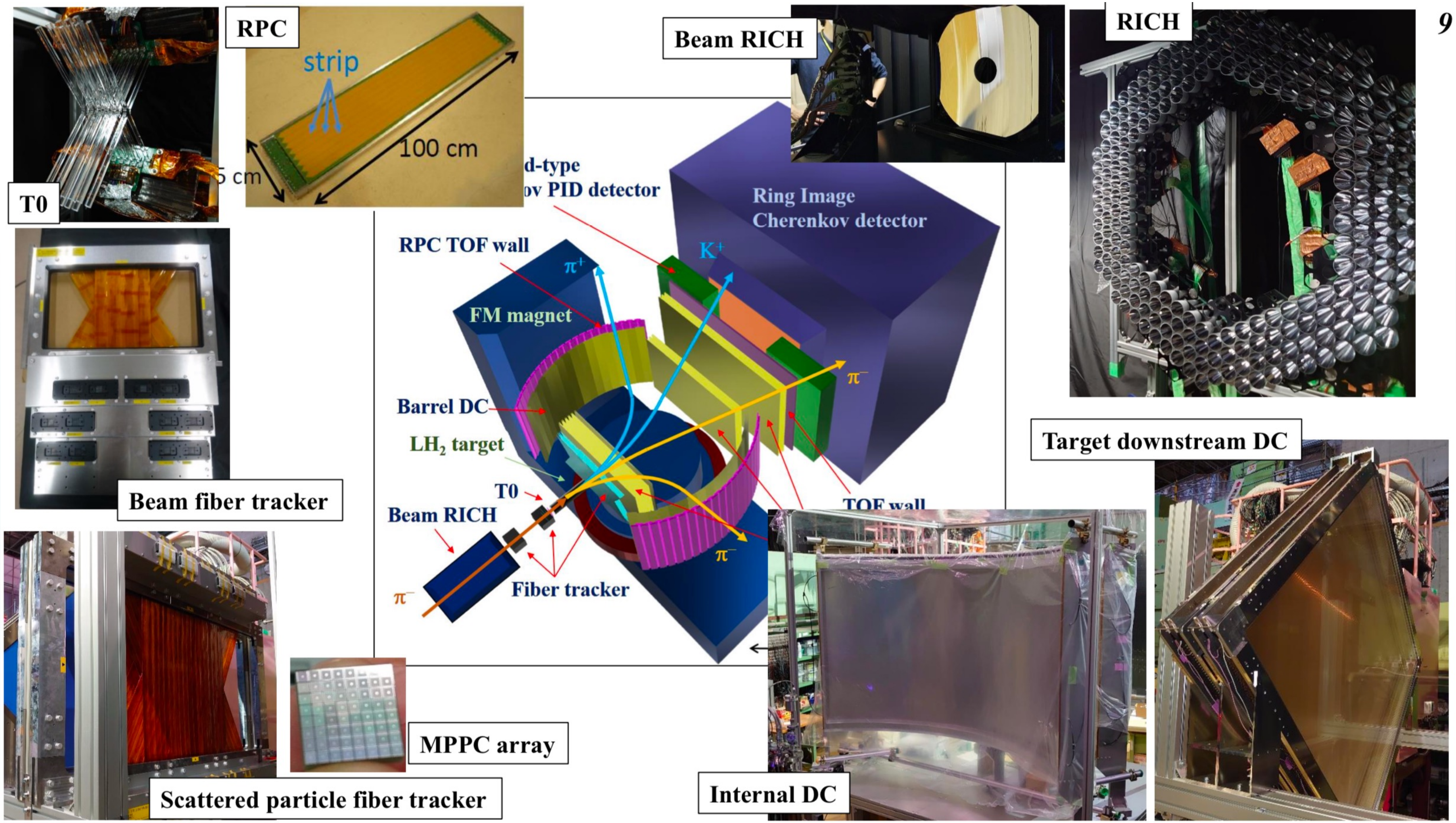
- Beam intensity ~30 MHz (60 M/2.0 sec. spill, 4.2 sec. cycle)

⇒ ~1.5 MHz reaction rate (4 g/cm^2 LH$_2$ target)

- Total expected data rate: ~13 GB/sec. (~100 Gbps during extraction)

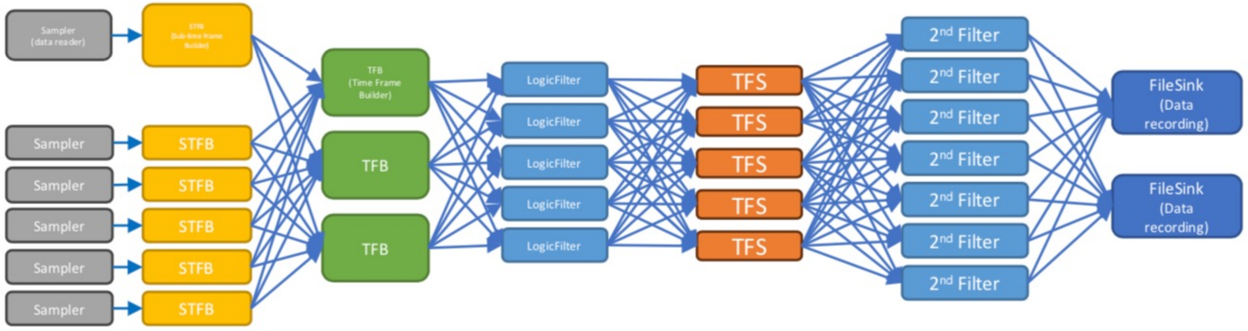
⇒ Reduction by filter: 1/100 ~ 1/1000

Recent tests at J-PARC

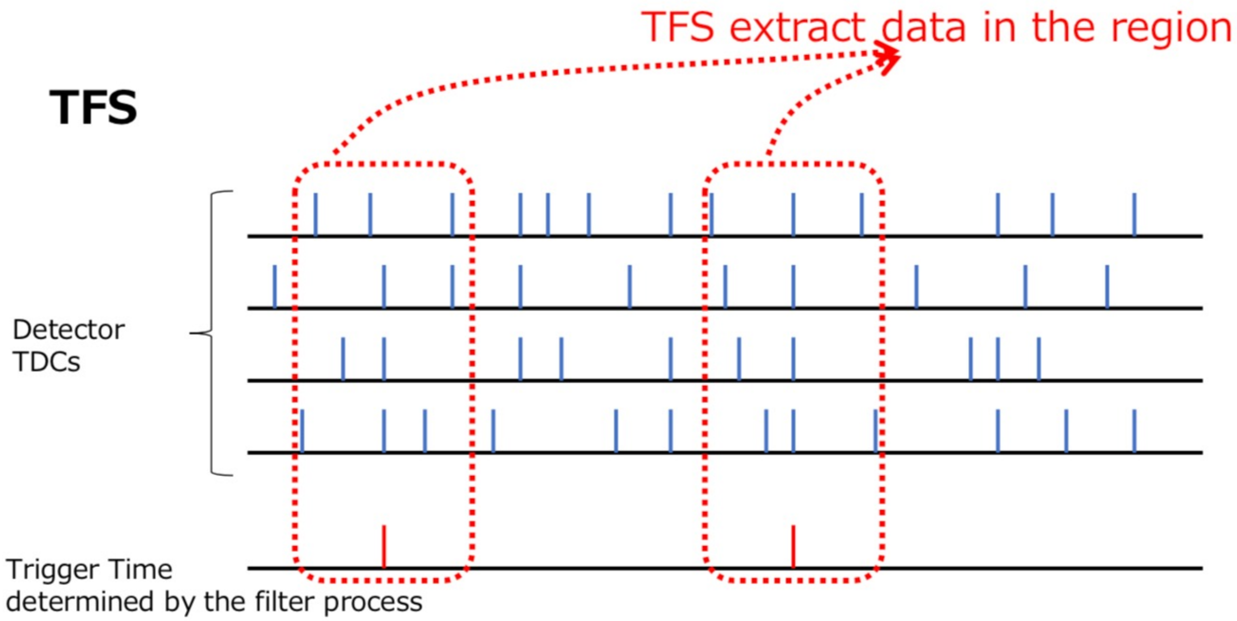


Recent tests at J-PARC

DAQ configuration



TFS



TFS extract data in the region

- Sampler
  - Reading data from front-end electronics
- Sub-Time Frame Builder (STFB)
  - Chopping the data from the sampler for each HBF, and several HBF are put together to make a Sub-Time Frame.
- Time Frame Builder (TFB)
  - Making Time Frame combined from Sub-Time Frame data from each Sub-Time Frame Builder
- Filter/Online software trigger
  - Finding the good event in the time frames.
- Time Frame Slicer (TFS)
  - Extracting the data in the time near the found trigger time.
  - EB reduces the size of data.
- 2nd level software trigger
  - Calculating TOF and extracting Kaon beam event.
- FileSink
  - Writing received data to the file.