DAQ System Development for CEPC

Xiaolu Ji

On behalf of CEPC DAQ group

2024.10.24

CEPC 2024 International Workshop

Outline

- Brief introduction to CEPC DAQ design
- Introduction to CEPC DAQ software framework
- How it work in JUNO DAQ
- Ongoing research activities for CEPC DAQ

CEPC Requirement – Data Rate

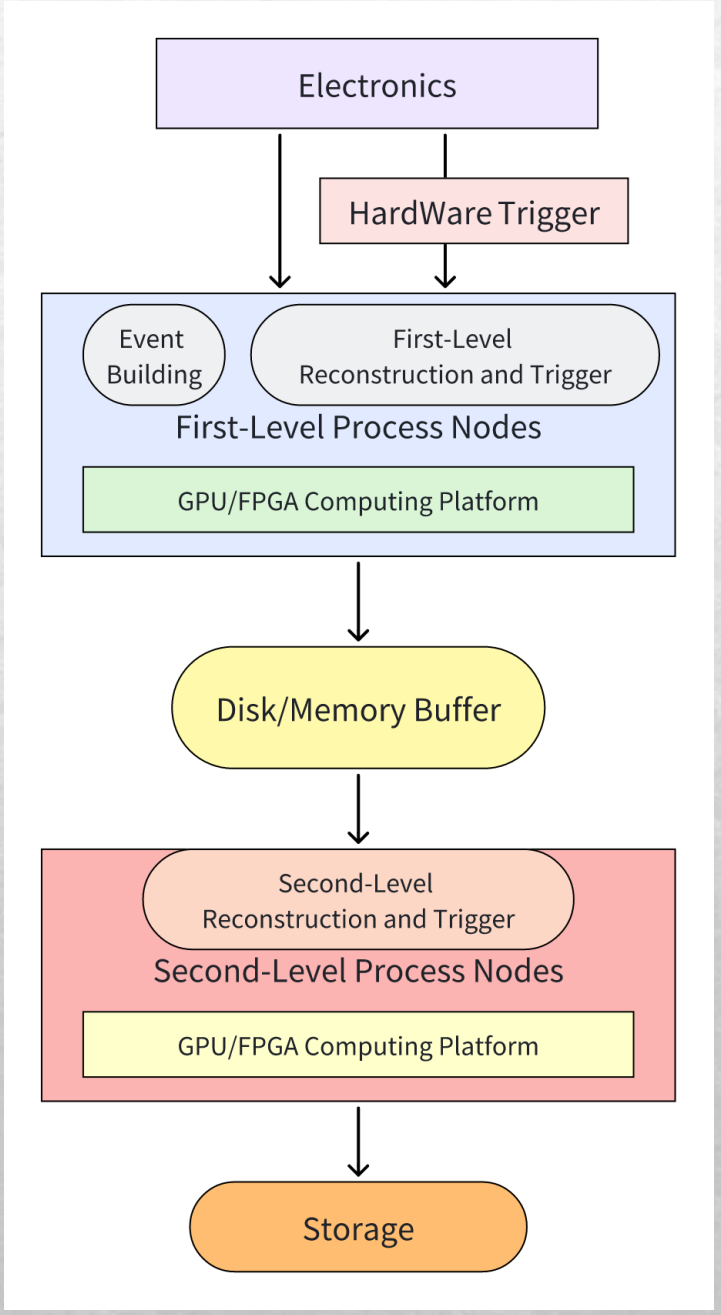
- **Data rate before trigger**
  - <1 TB/s @ Higgs
  - Several TB/s @ Z
- L1 trigger rate
  - *O*(1 k) Hz @ Higgs
  - *O*(100 k) Hz @ Z
- Event size < 2 MB
- Storage rate after HLT
  - <100 Hz(200 MB/s) @ Higgs
  - 100 kHz (200 GB/s) @ Z

Preliminary background and data rate estimation

	Vertex	Pix(ITKB)	Strip (ITKE)	OTKB	OTKE	TPC	ECAL-B	ECAL-E	HCAL-B	HCAL-E	Muon
Channels per chip	512*1024	512*128	1024	128		128	8~16				
Data Width /hit	32bit	42bit	32bit	48bit		48bit	48bit				
Avg. data rate / chip	0.18Gbps/chip, 1Gbps/chip inner	3.53Mbps/chip	21.5Mbps/chip	2.9Mbps/chip	38.8Mbps/chip	~70Mbps/module Inmost	10kHz/ch	10kHz/ch	5kHz/channel	5kHz/channel	10kHz/channel, 20kHz/iner endcap
Detector Channel/module	1882 chips @Stch &Ladder	30,856 chips 2204 modules	23008 chips 1696 modules	83160 chips 3780 modules	11520 chips 720 modules	492 Module	0.96M chn ~60000 chips 480 modules	0.39 M chn	3.38M chn 5536 aggregation board	2.24M chn 1536 Aggregation board	43,176 chn(iner end-cap 6912), 288 modules
Avg Data Vol before trigger	474.2Gbps	101.7Gbps	298.8Gbps	249.1 Gbps	27.9Gbps	34.4Gbps	460.8Gbps	187Gbps	811.2Gbps	537.6Gbps	24Gbps
Occupancy	0.22e-4	2.5e-4				2.8e-4	58e-4			19.5e-4	
Sum	3.2 Tbps = 400 GB/s @Higgs										

Architecture Design of CEPC DAQ

- **Compatible design with or without HW trigger**
- Full COTS(commercial-off-the-shelf) hardware
- Readout interface and protocol
  - Ethernet 100Gbps, TCP or RDMA based
- Use RADAR software framework
- GPU/FPGA for processing acceleration
- Disk or memory buffer
  - Decouple computing environments
  - Offline algorithm can be easily integrated online



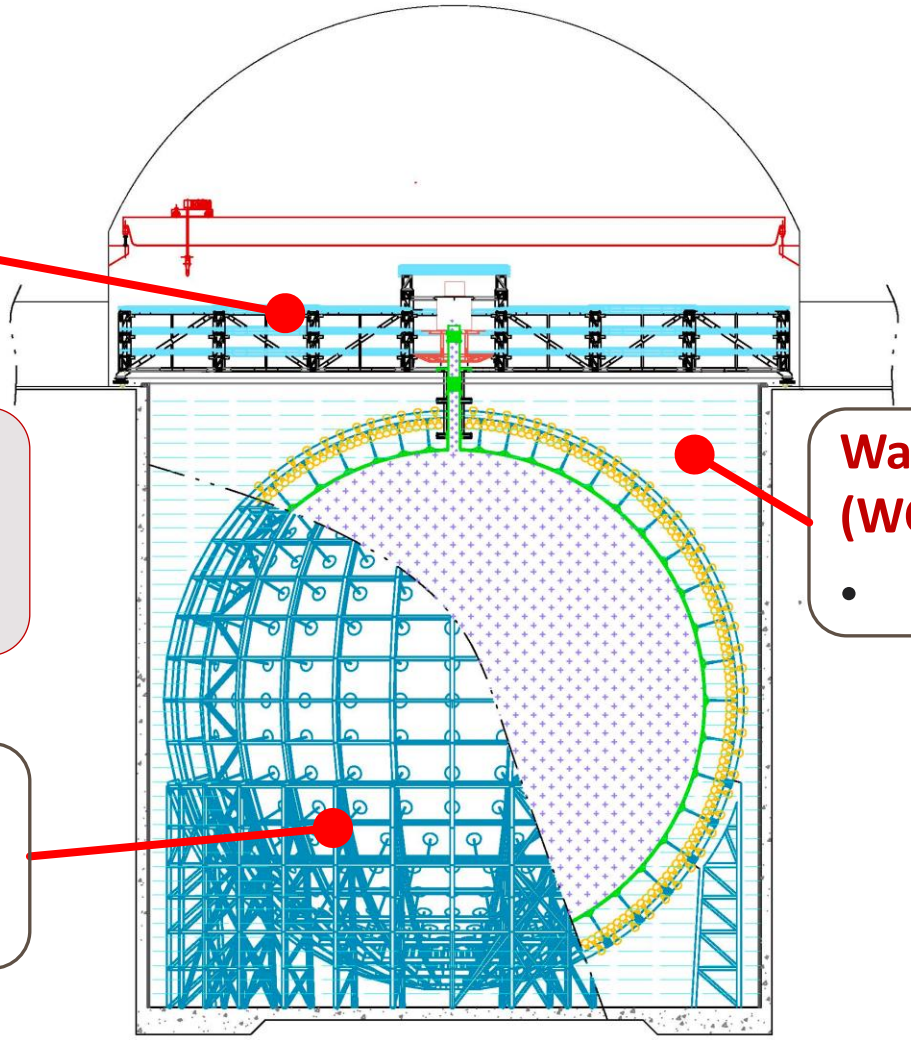
Streaming Readout Framework – Radar

*hete**R**ogeneous **A**rchitecture of **D**ata **A**cquisition and p**R**ocessing*

- **V1:** deployed in LHAASO (~ 5 GB/s data rate), software trigger mode
- **V2:** upgraded for JUNO (~ 50 GB/s data rate), mix trigger mode
  - Containerized running
  - Partial High Availability
- **V3:** CEPC-oriented (~ TB/s data rate) , under development
  - High-throughput data transfer and processing
  - Heterogeneous online processing platforms

JUNO DAQ Highlights

Top Tracker (TT)



LPMT: 1 GS/s waveform readout with global trigger + trigger-less T/Q data

SPMT: trigger-less readout of T/Q

Central Detector (CD)

- 17,612 20-inch PMTs (LPMT) and 25,600 3-inch PMTs (SPMT)

Water Cherenkov Detector (WCD)

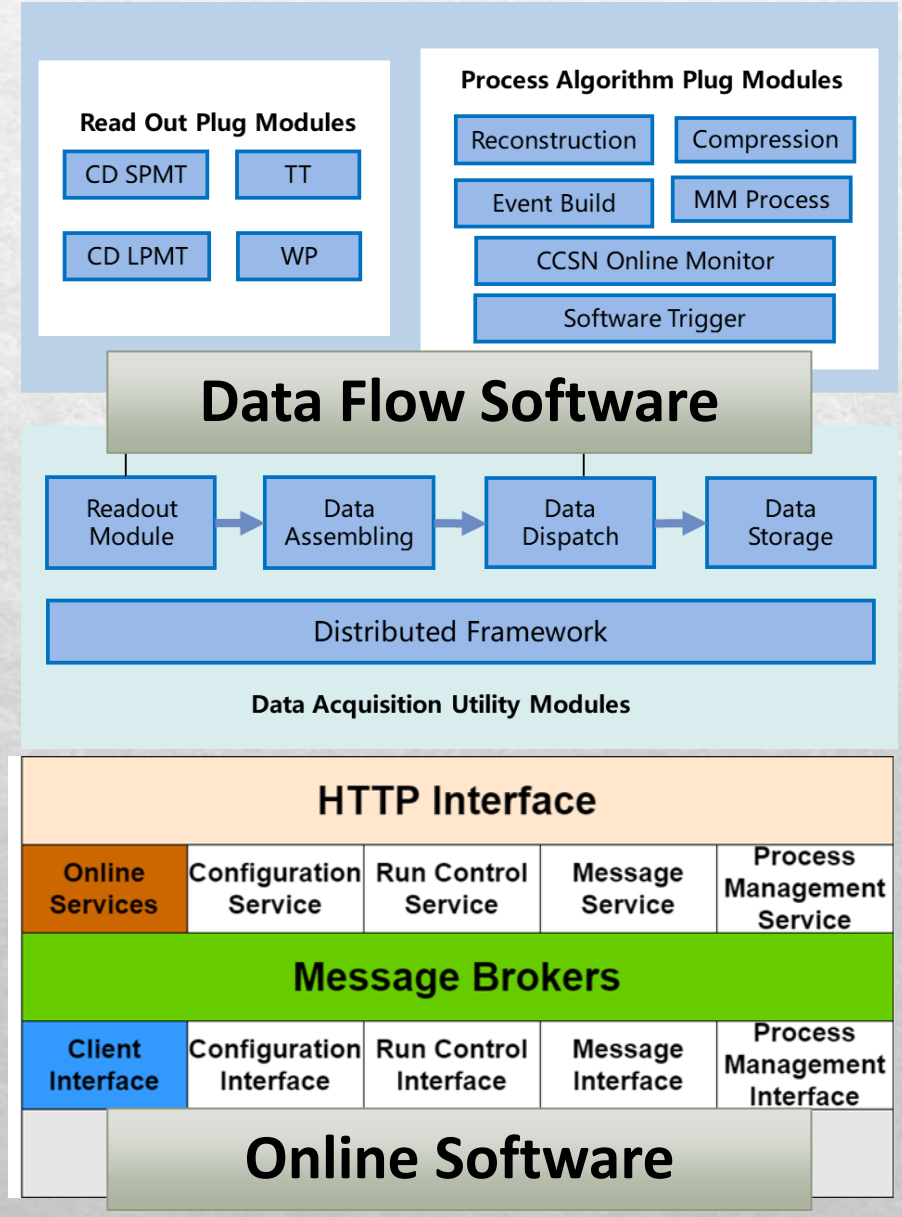
- 2,400 20-inch PMTs

LPMT readout

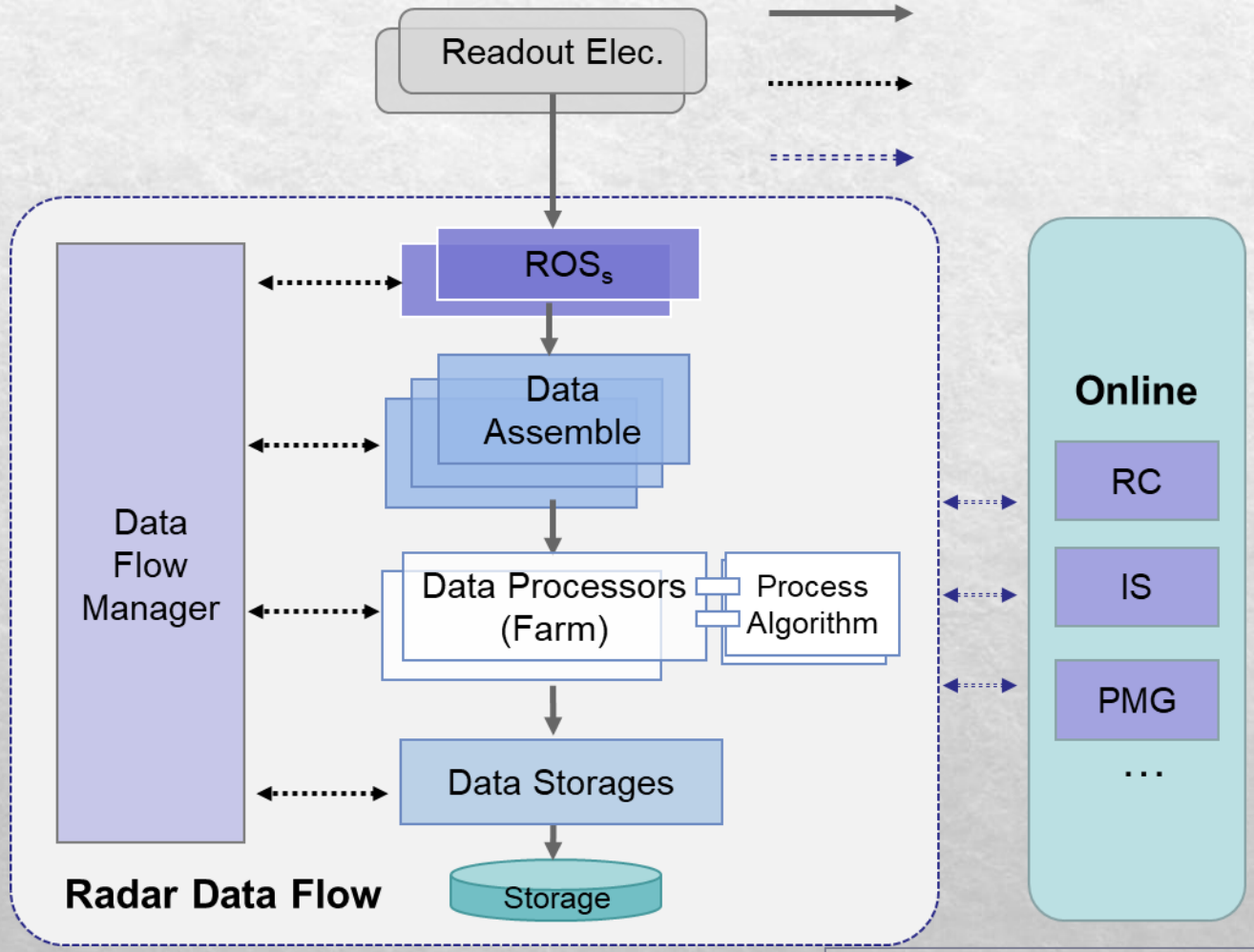
- **> 40 GByte/s** triggered waveform data and trigger-less time and charge data
- **~ 7000 readout links** with interface: 1 Gbps Ethernet + TCP protocol
- Process events via **Online Event Classification** to reduce data rate by **~ 500 times**

JUNO DAQ Software Architecture

- **Radar**
  - V2: upgraded for JUNO
- **General-purpose distributed framework**
  - Transport layer – ZeroMQ
  - Services – Kafka / ZooKeeper based
- Divided into two parts:
  - **Data flow software:** process data streams
  - **Online software:** management and services



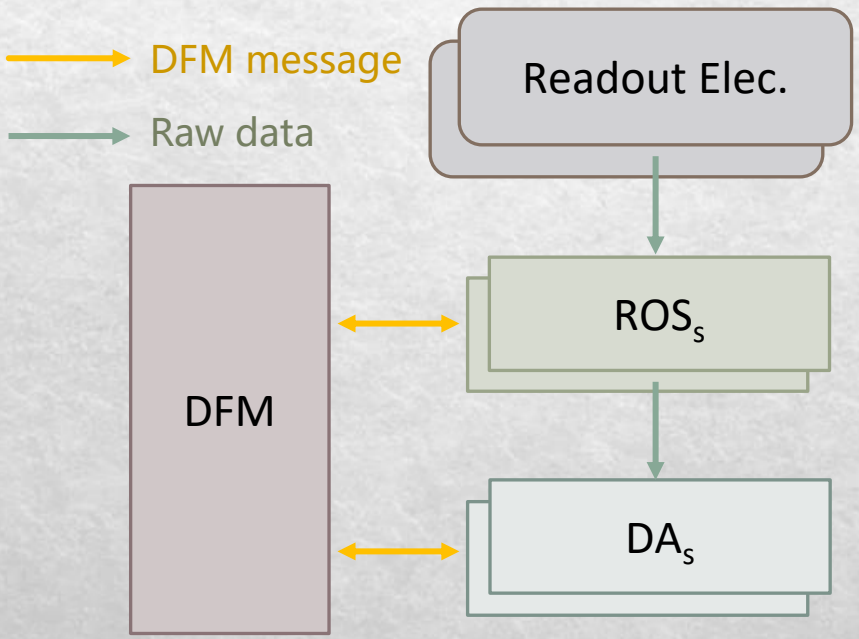
Data Flow Software



- Lightweight structure
- ROS + DA + DP + DS
- **Plug-in modules design** for ROS & DP
- Integrate customized readout / processing modules

- **ROS:** ReadOut System
- **DA:** Data Assemble
- **DP:** Data Processor
- **DS:** Data storage

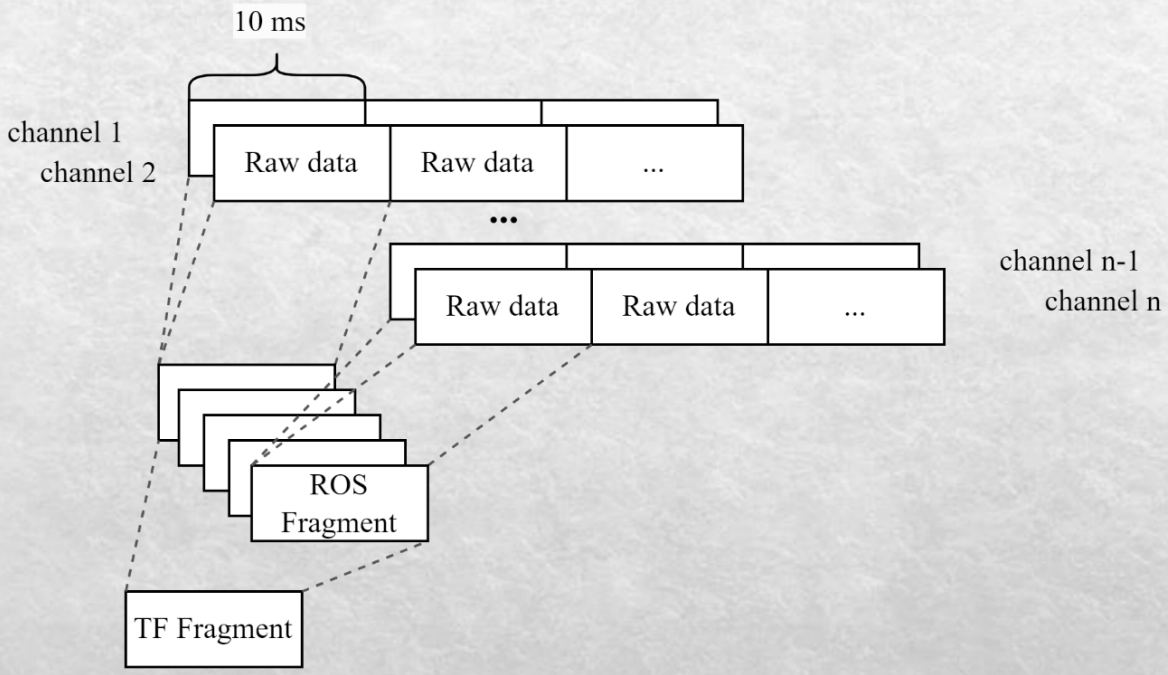
JUNO DAQ Data Assemble



Raw data stream

ROS Fragments

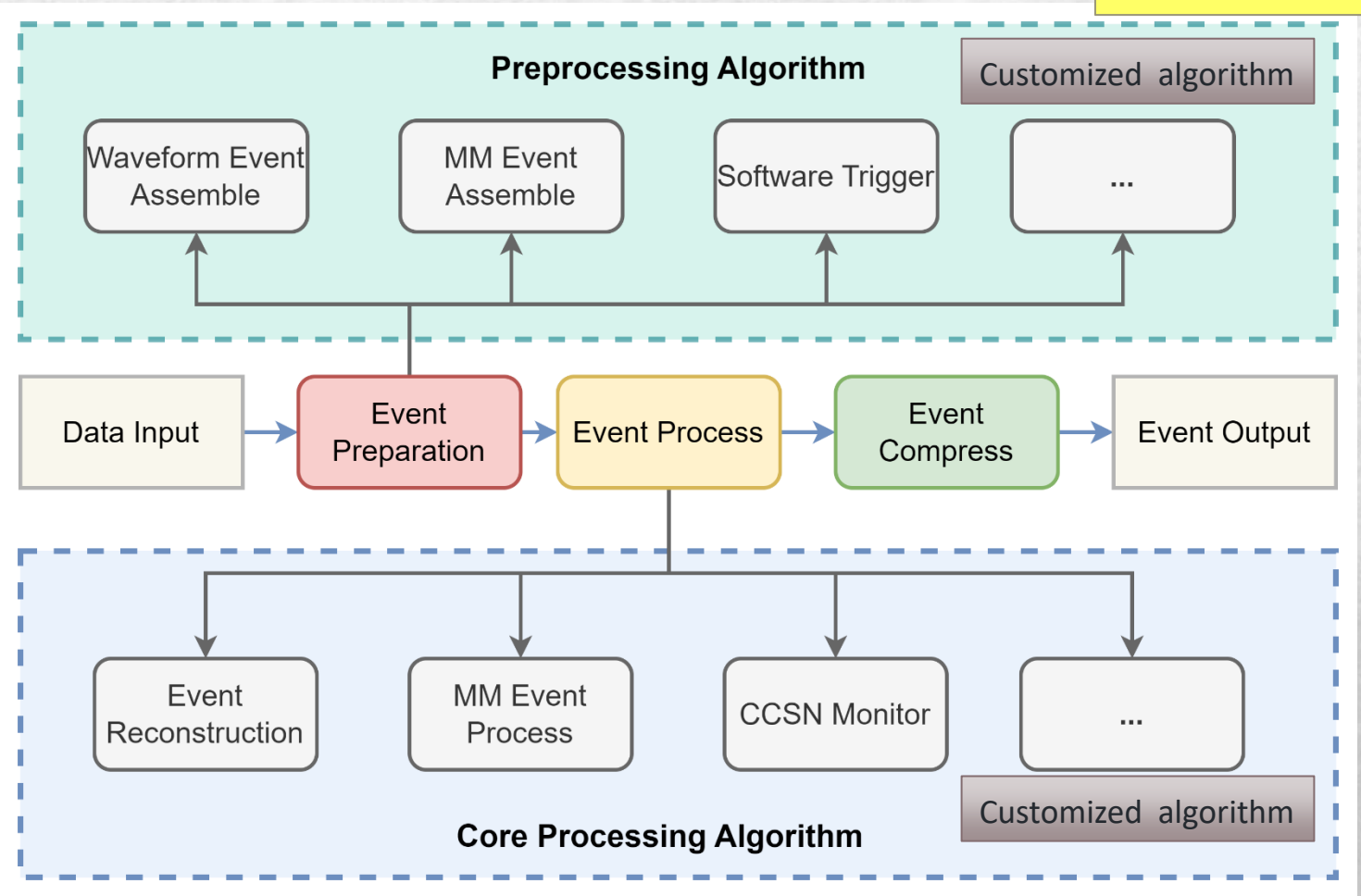
Timed Fragment



- Support data assembled by trigger ID or timestamp
- Uniform processing both triggered and trigger-less data
- **2 level assemble** by time fragments – ROS + DA
  - Based on global clock system provided by WR

JUNO DAQ Data Processing

Data processing is flexible, configurable



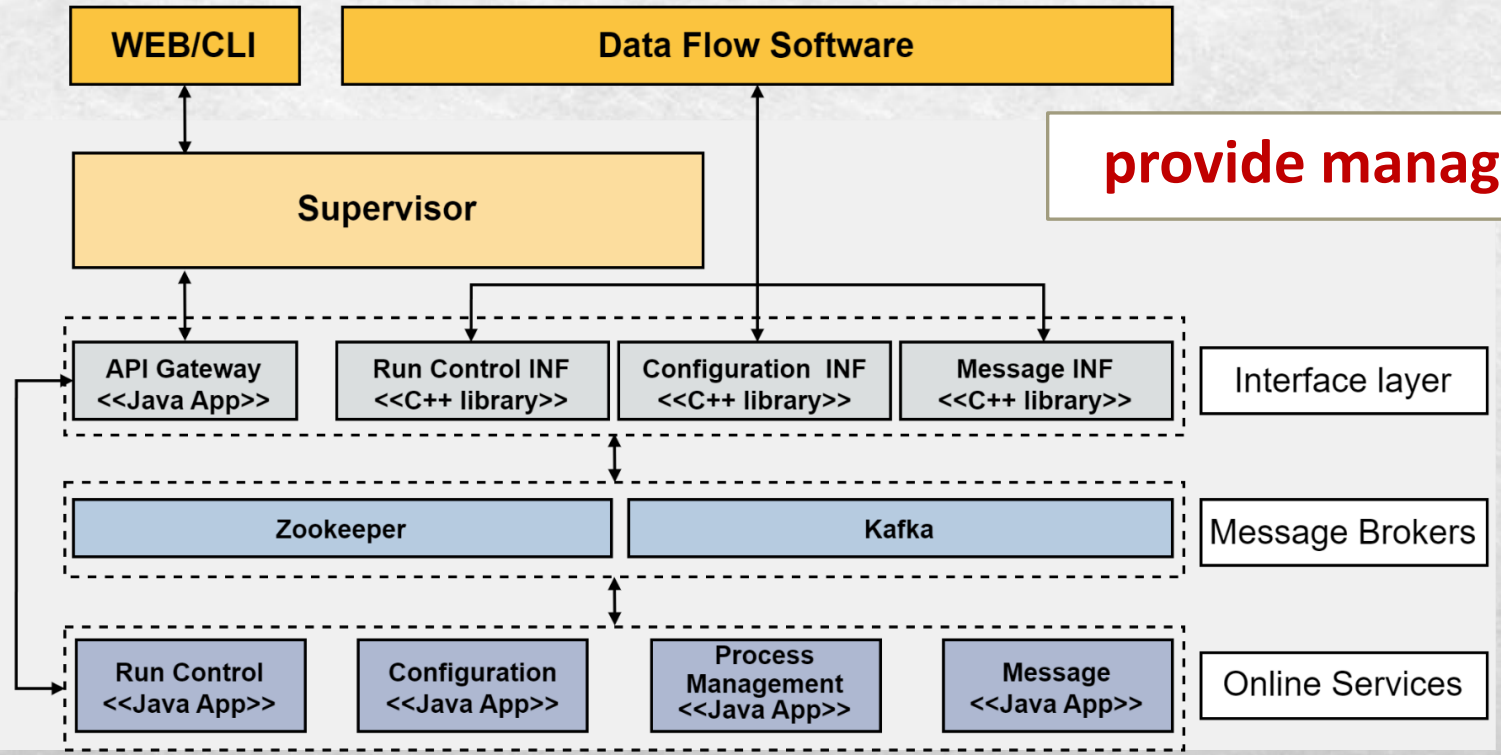
2-stage data processing:

- Data preparation / preprocessing
- Core event processing algorithms

- Support parallel processing
- **Processing interface provided**
- **Plugin algorithm** deployment

- **CCSN:** Core Collapse SuperNova
- **MM:** Multi-Messenger

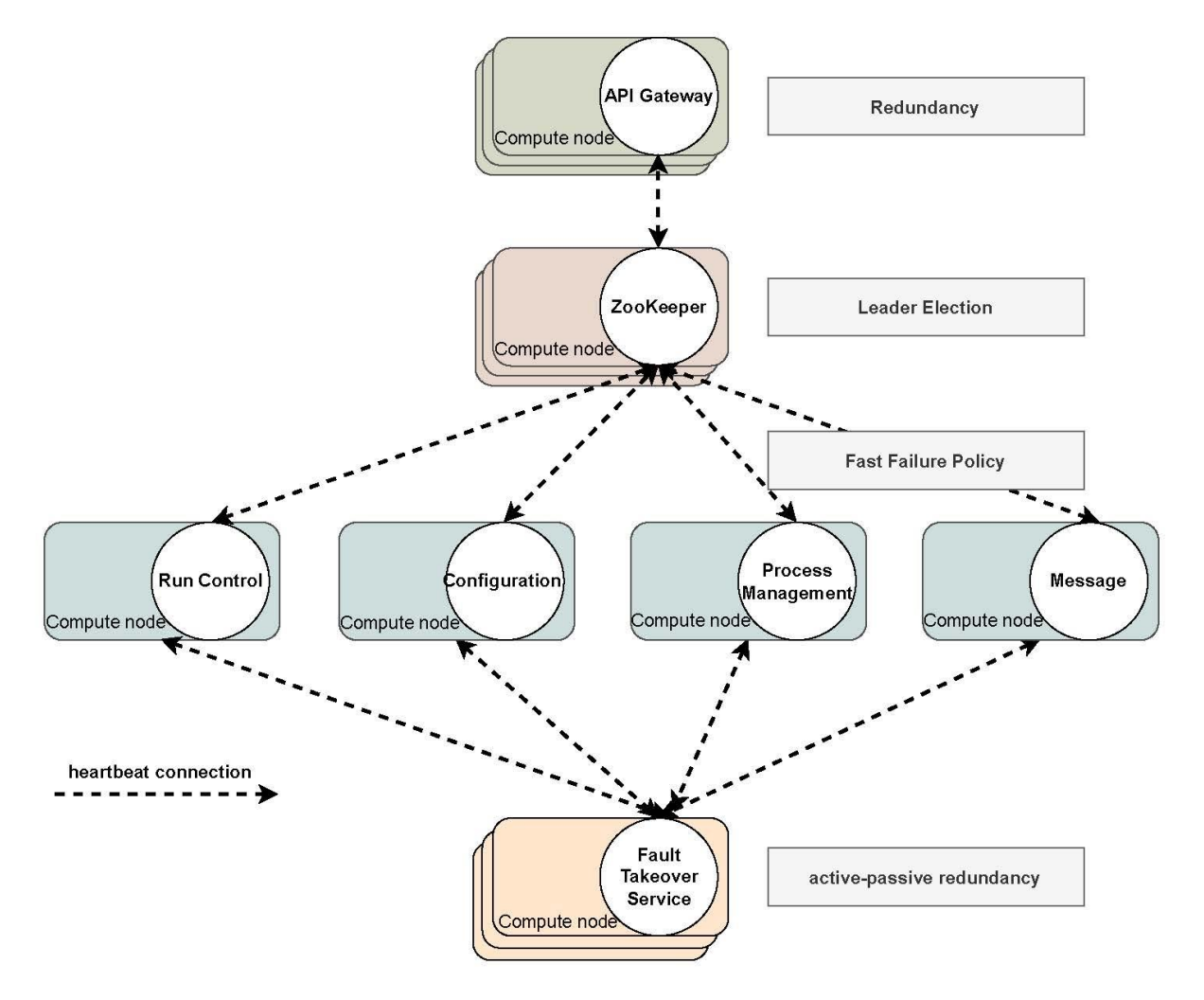
Online Software Overview



provide management, interfaces and services

- **Centralized messaging topology:** message brokers – decouple online and data flow
- **Microservices architecture**: keep independence between services
- Layered design – interface, message, online services and supervisor
- Kubernetes managed containerized operation: support 30 years running for JUNO lifetime

Online HA Design & Implementation



Failure Detection

Heartbeat Detection

- Inter-Process: Based on ZooKeeper
- Inter-Node: Based on ICMP Protocol

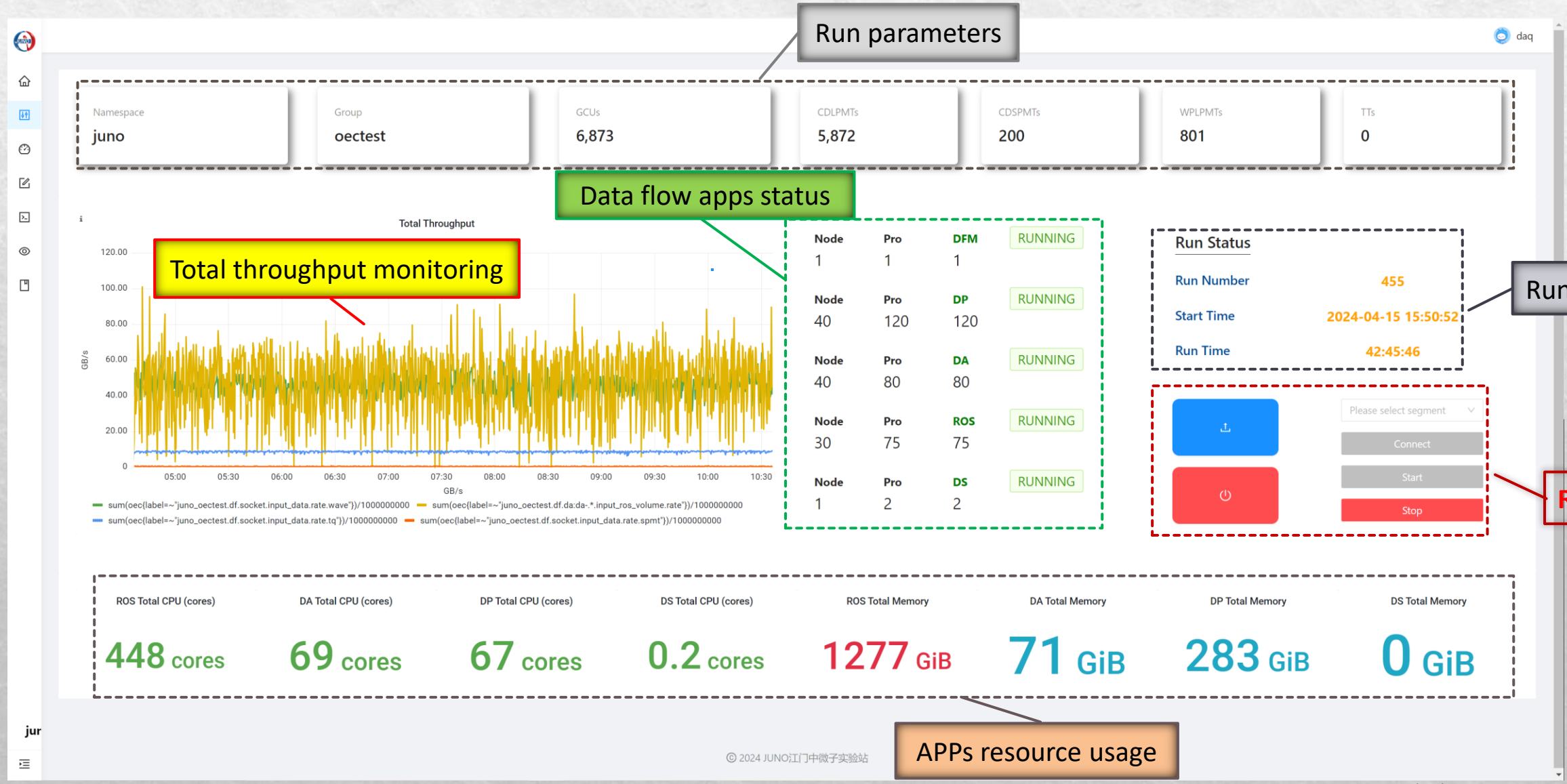
Failover

- Redundancy + Master-Slave Election
- Takeover scheme based on Kubernetes
- Solutions for rapid restart of services

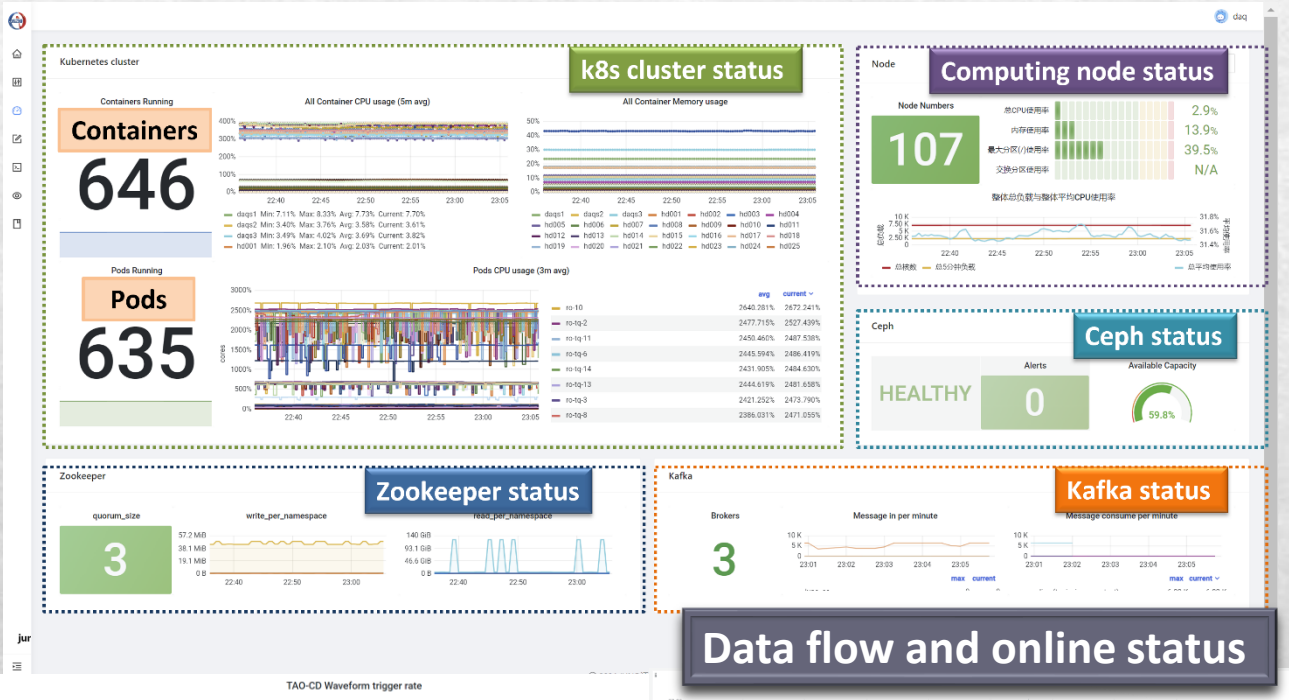
Reliable for supernova detection

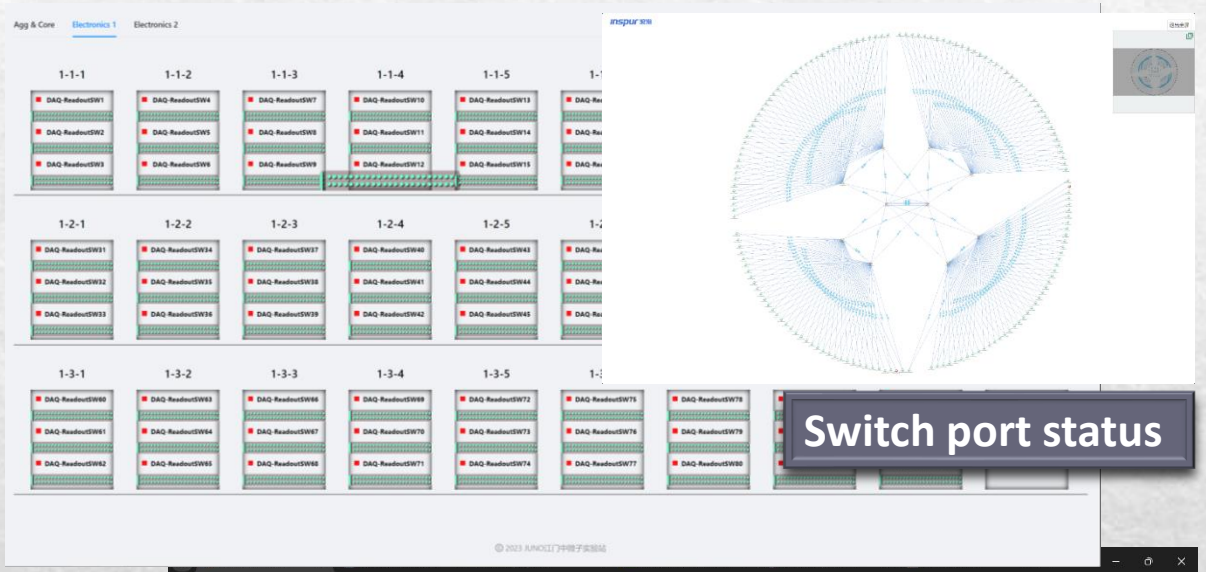
Anomaly detection and recovery have been fully considered to ensure system reliability

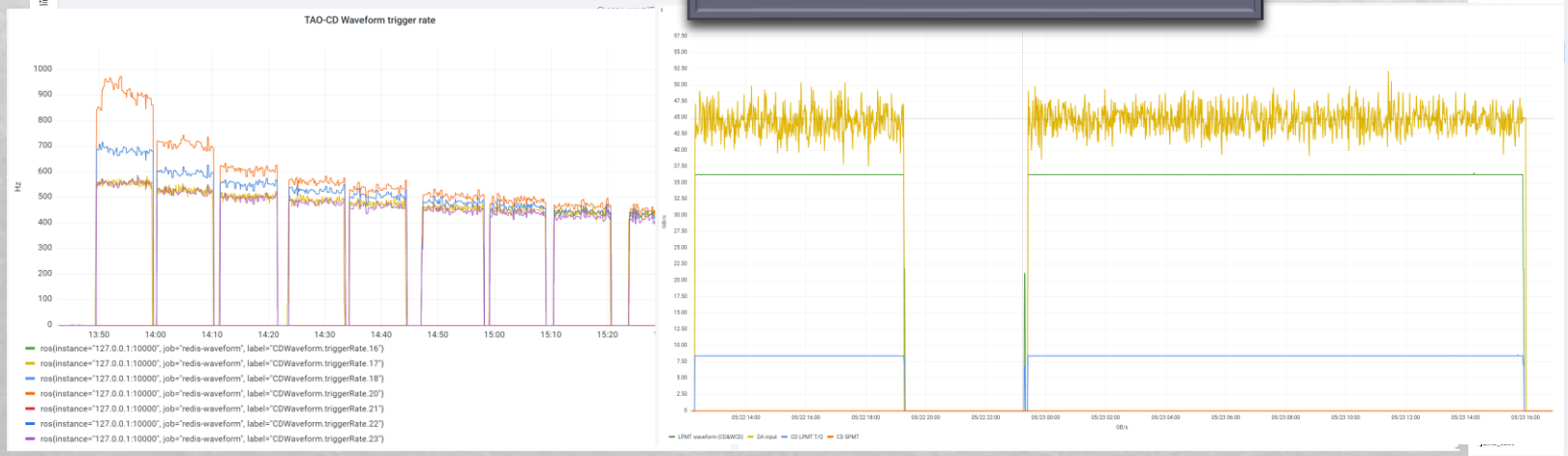
JUNO DAQ Console GUI

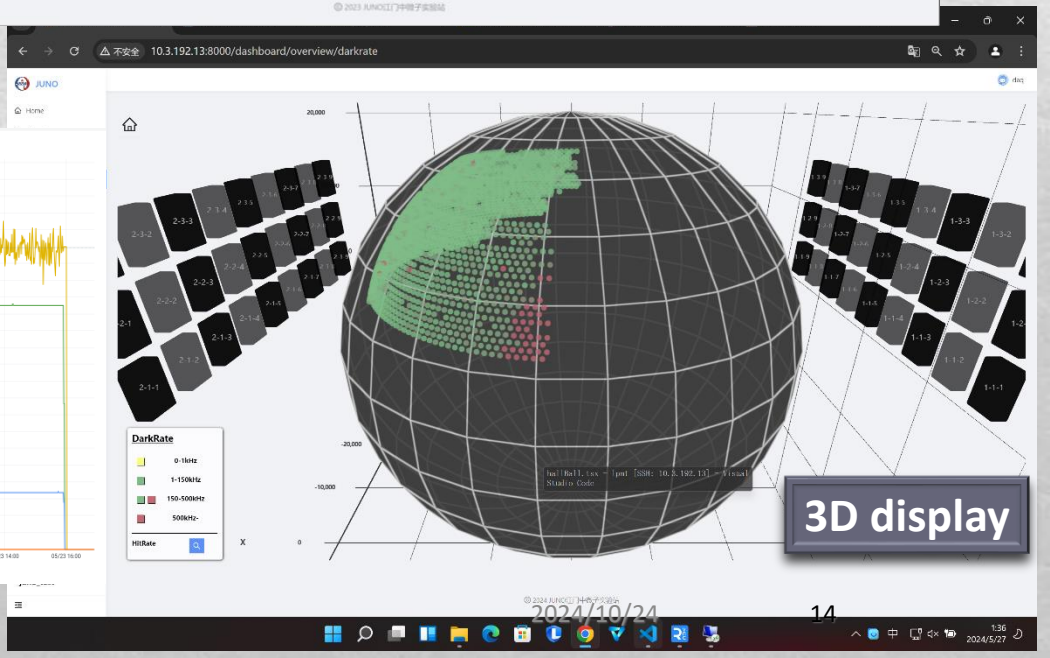


Visualization Monitoring

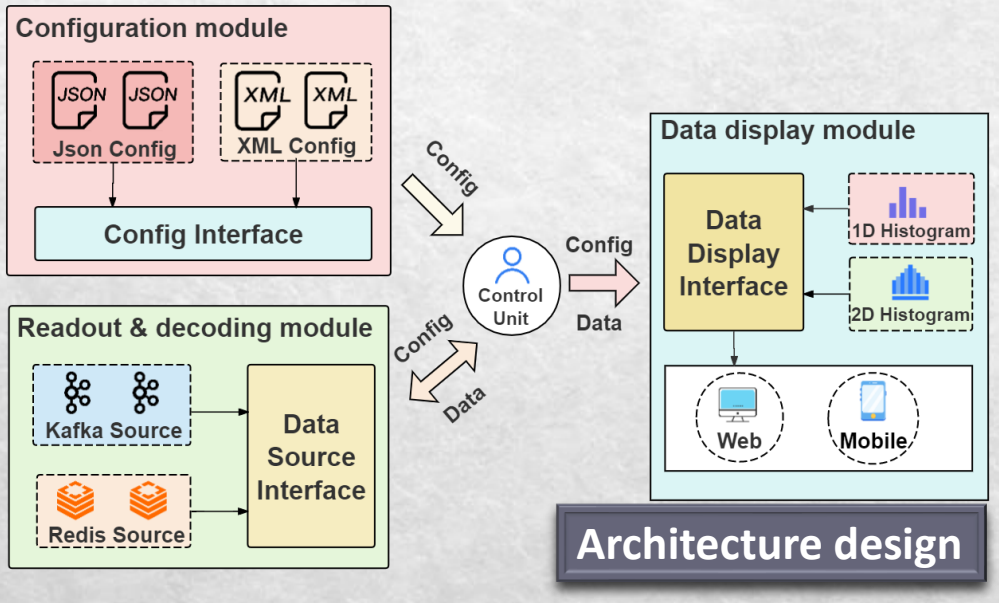




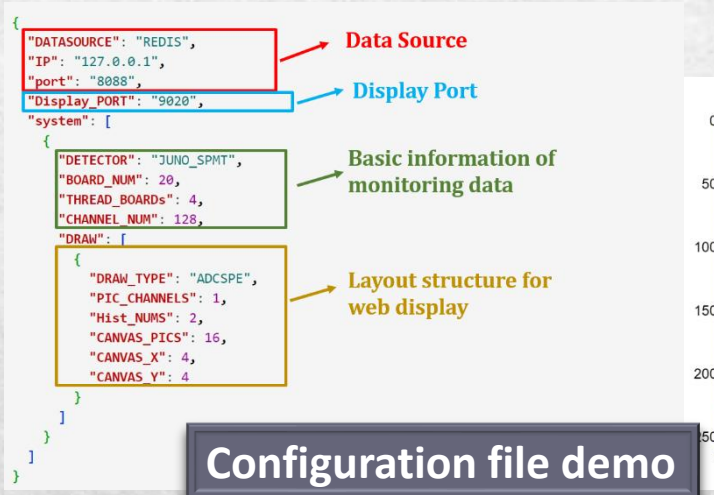




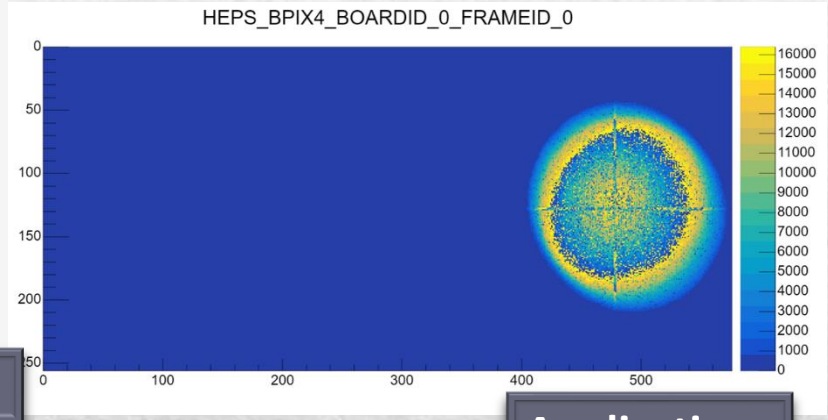
ROOT-based Online Data Visualization System (ROBOT)



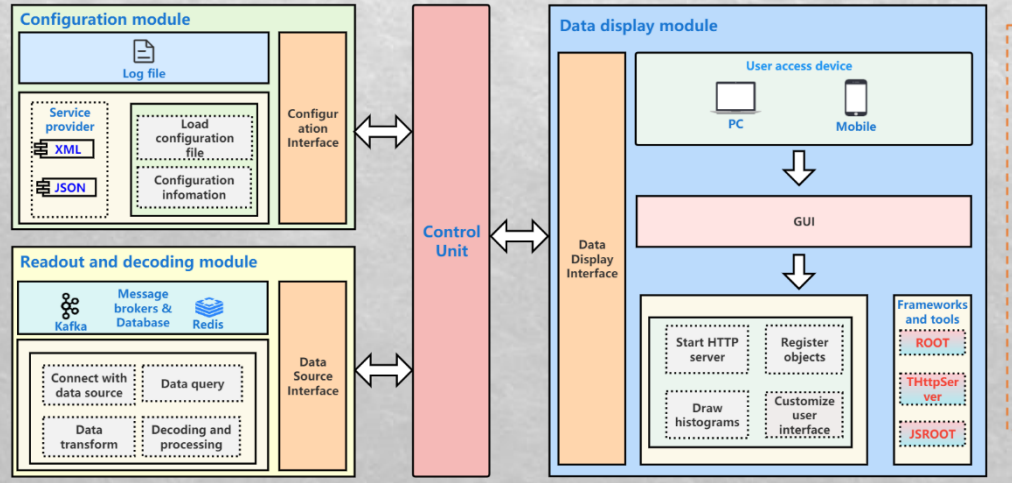
Architecture design



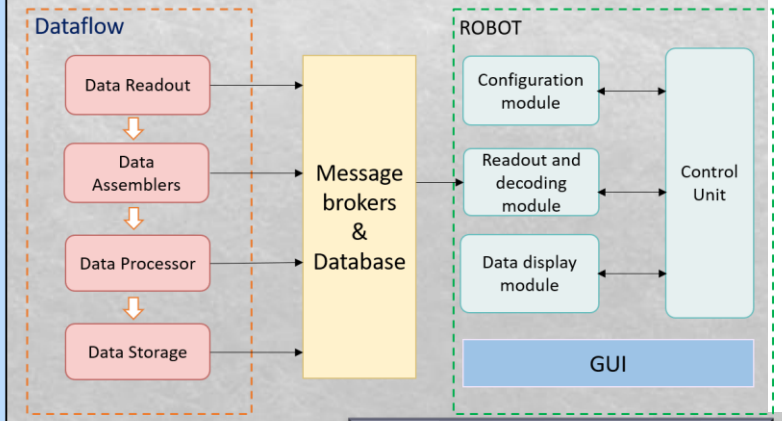
Configuration file demo



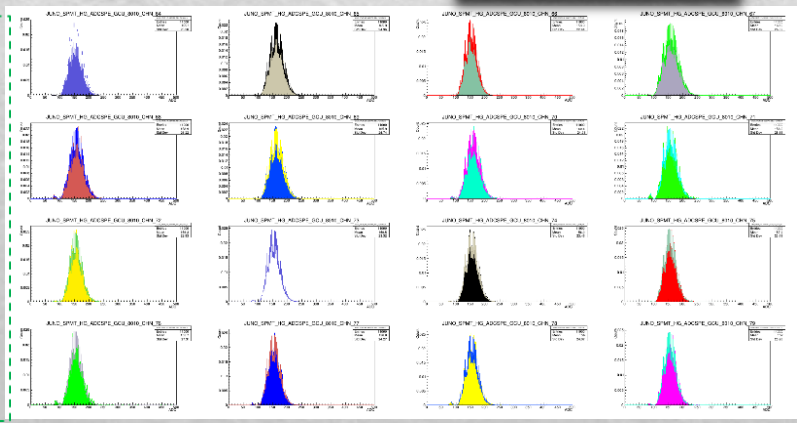
Applications



System implementation



Integration scheme



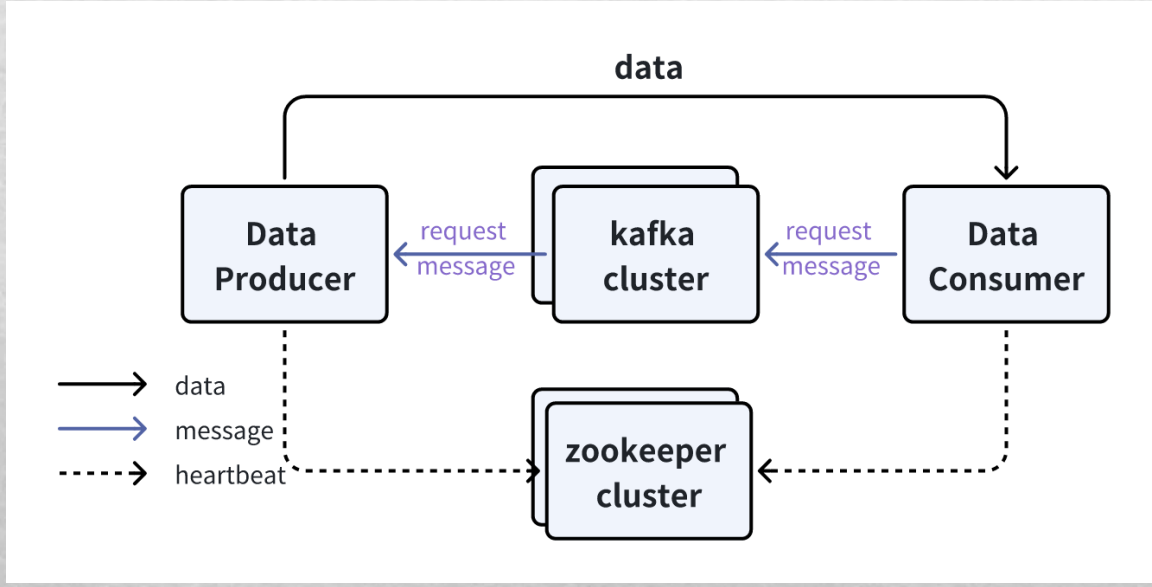
Next – Current R&D Progress

- Dataflow HA Design
- Online Processing Strategy and computing Accelerating
- High-throughput Distributed Memory Cache Pool
- RDMA Research
- AI-based Utilities Research

Dataflow HA Design

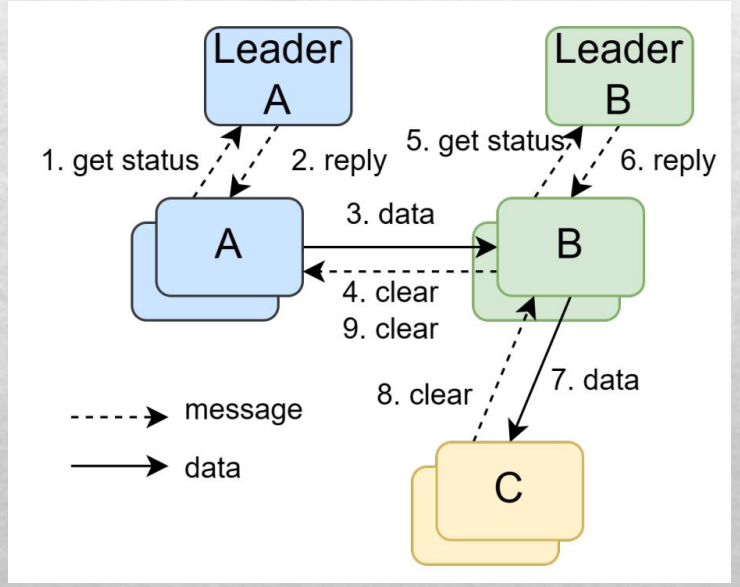
- Goal: Automatically recover without data loss when an exception occurs
- Trying different solutions

Utilize high-availability middleware (Kafka)



- Data recovered from backup nodes

Develop new data recovery components



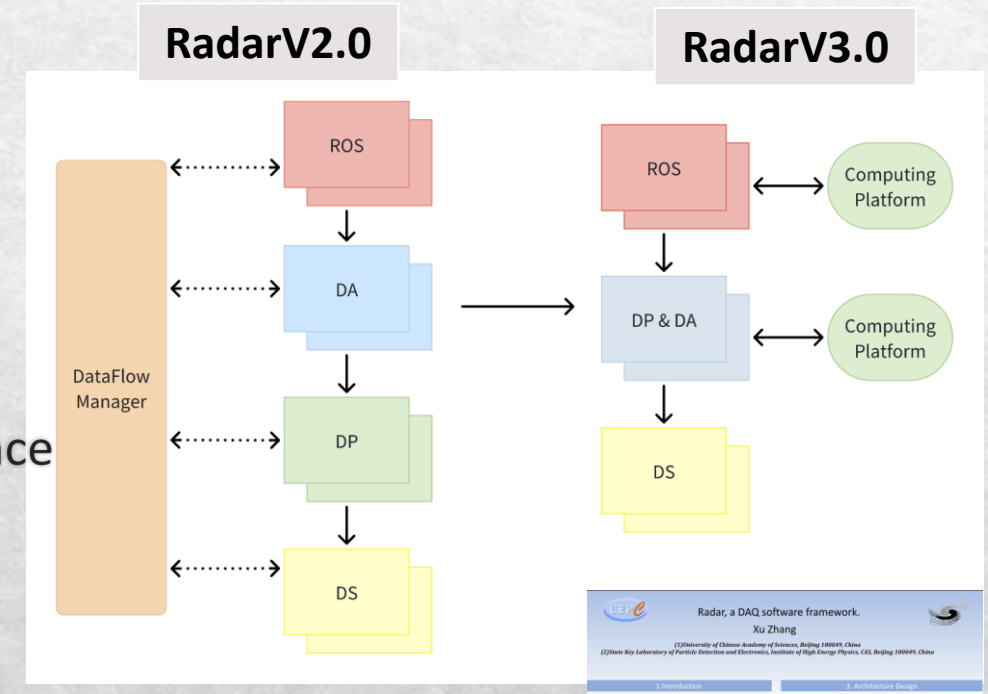
- Data recovered from upstream node

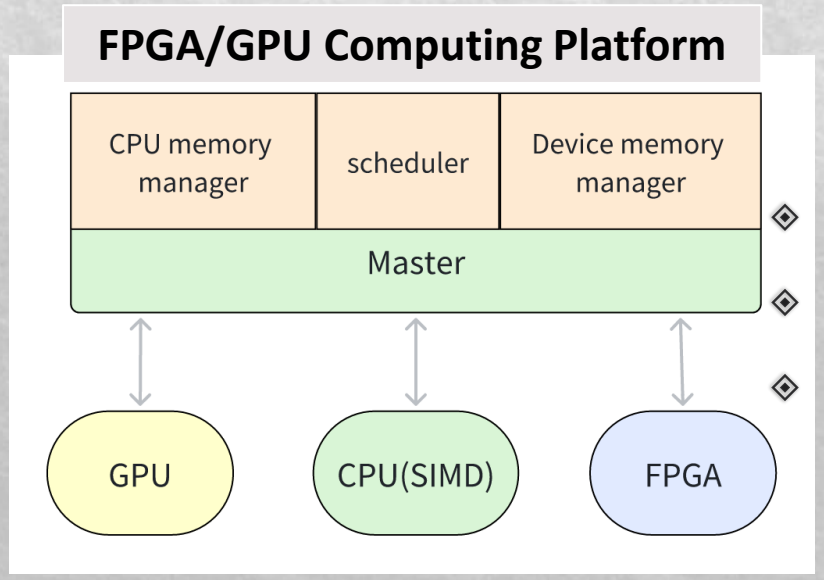
Testing and comparing the performance, availability, and flexibility

Online Processing Strategy

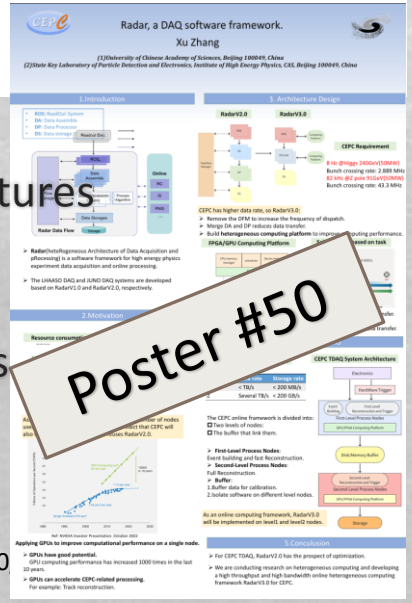
CEPC has higher event & data rate requirement

- Remove DFM to increase the maximum dispatch frequency
- Merge DA and DP to reduce data transfer
- Build heterogeneous computing platform to improve performance





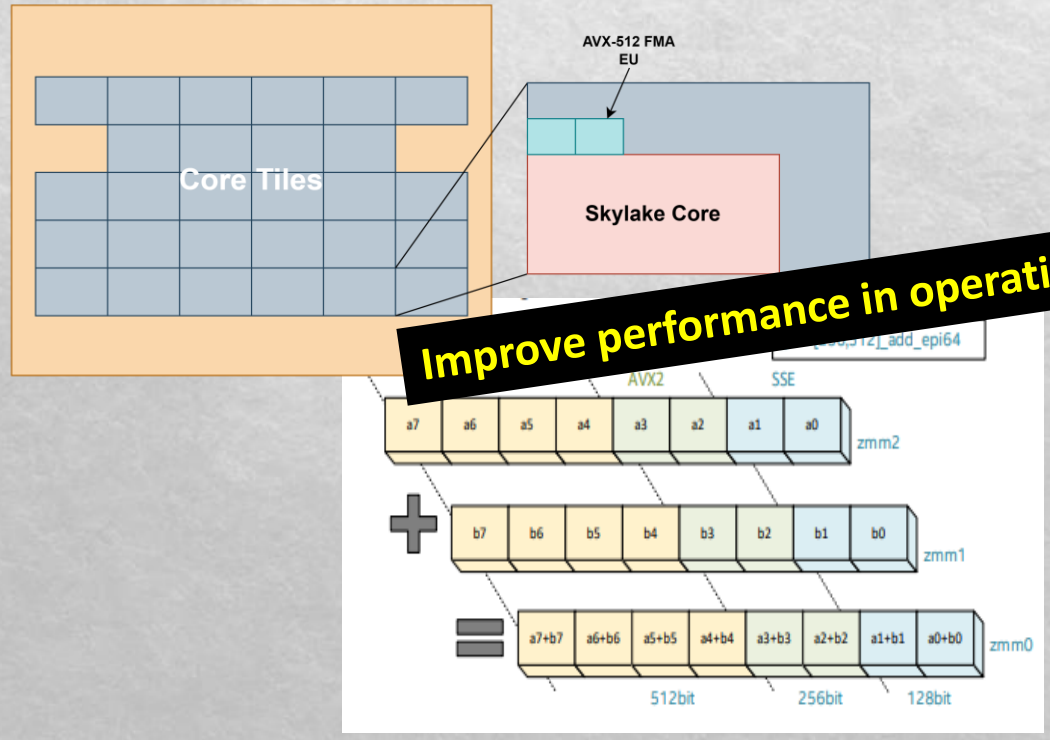
- CPU Memory Manager: Organize parallel computing data structures
- Scheduler: Assign algorithms to devices, manage data transfer
- Device Memory Manager: better scheduling memory resources

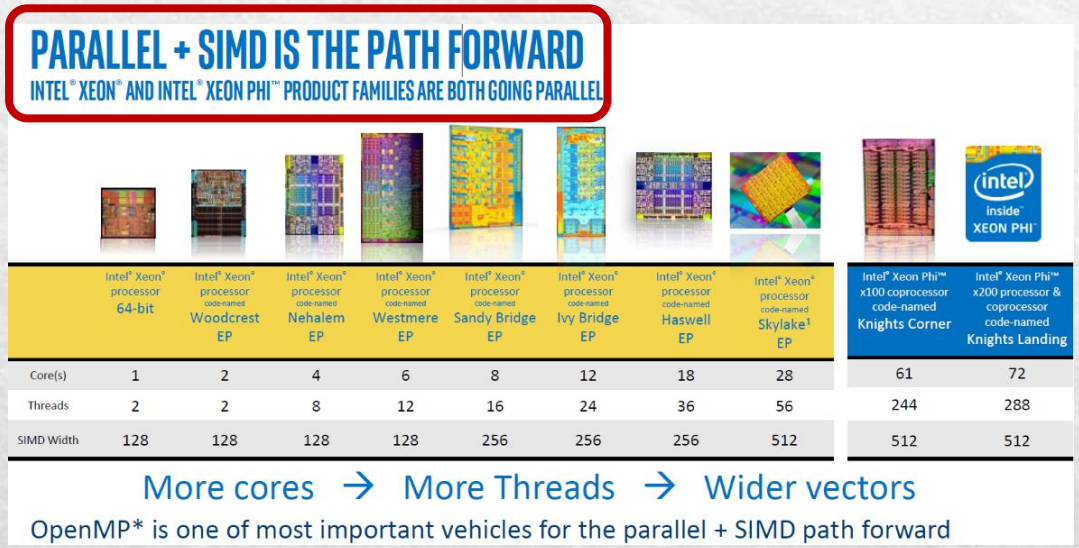


CPU Accelerating

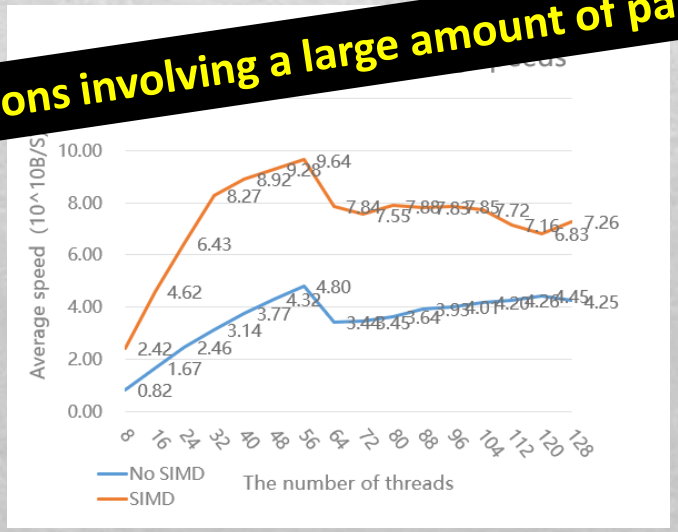
- **SIMD**(Single Instructions Multiple Data Parallel Processing)
  - — single controller managing multiple processors
- **OpenMP**(Open Multi-Processing)
  - — Parallel technology model
- **High-performance, flexible, user-friendly, portable**

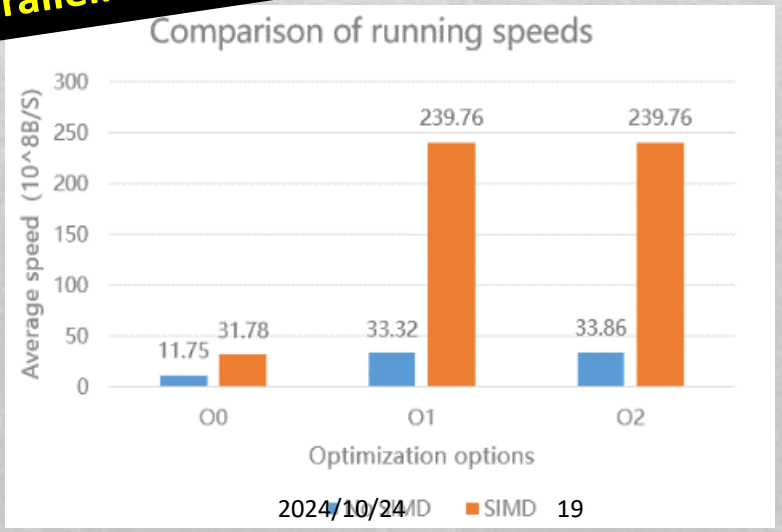
Intel Skylake Processor Architecture





Improve performance in operations involving a large amount of parallelizable data

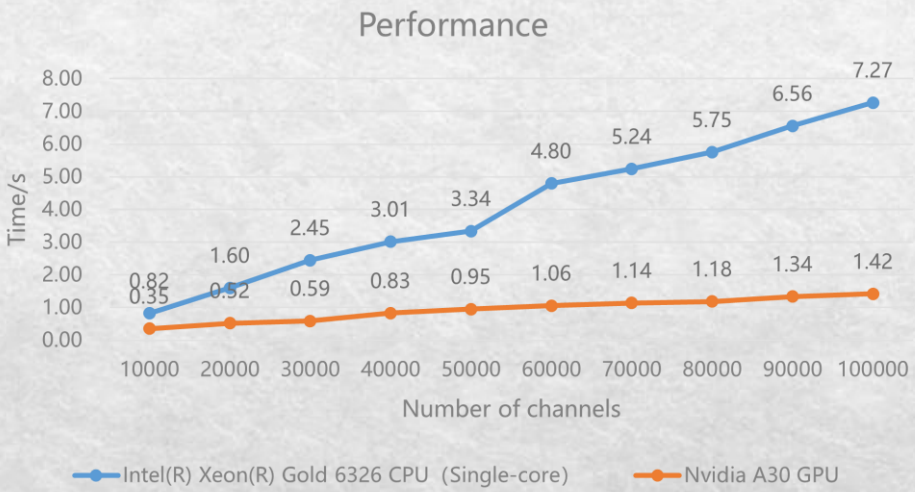




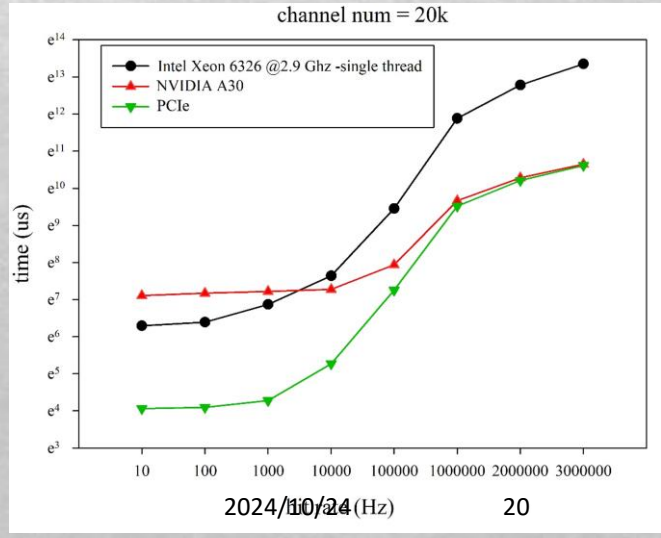
GPU Accelerating

- Implement some algorithm demo modules
- Focus on performance research
  - Batching, Data Parallelism, Memory Optimization, Asynchronous Computation, Multi-streaming…
- GPU can achieve impressively performance when the algorithms are suitable
- PCIe transfer bottlenecks cannot be ignored
  - complete as many algorithms as possible in a single transfer → maximize GPU resource utilization and alleviate transfer pressure
- Next: study acceleration strategies with the experimental scenario

GPU based online waveform reconstruction in JUNO

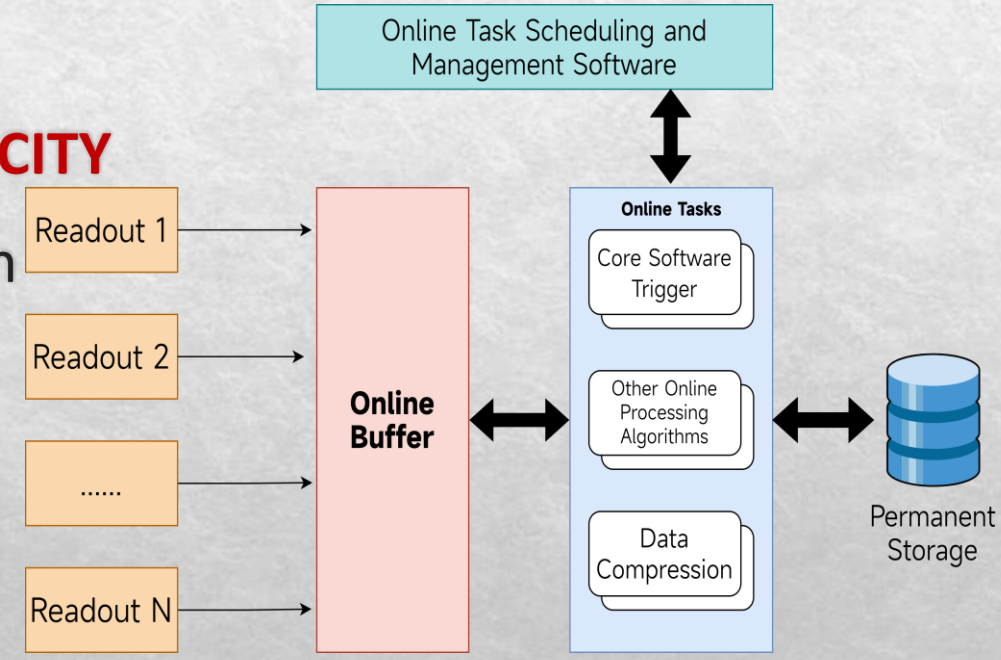


GPU based nHit trigger algorithm study



High-throughput Distributed Memory Cache Pool

- Decouple different data flow stages
- **Memory** speed is much higher than **Disk**
- Online computing: **more THROUGHPUT than CAPACITY**
- Memory Aggregation to provide IO interface with
- High throughput capability
- High operational efficiency
- High reliability



- Trade-off among overall performance, availability, and cost
- OR or AND buffer strategy

Research Status

Progresses

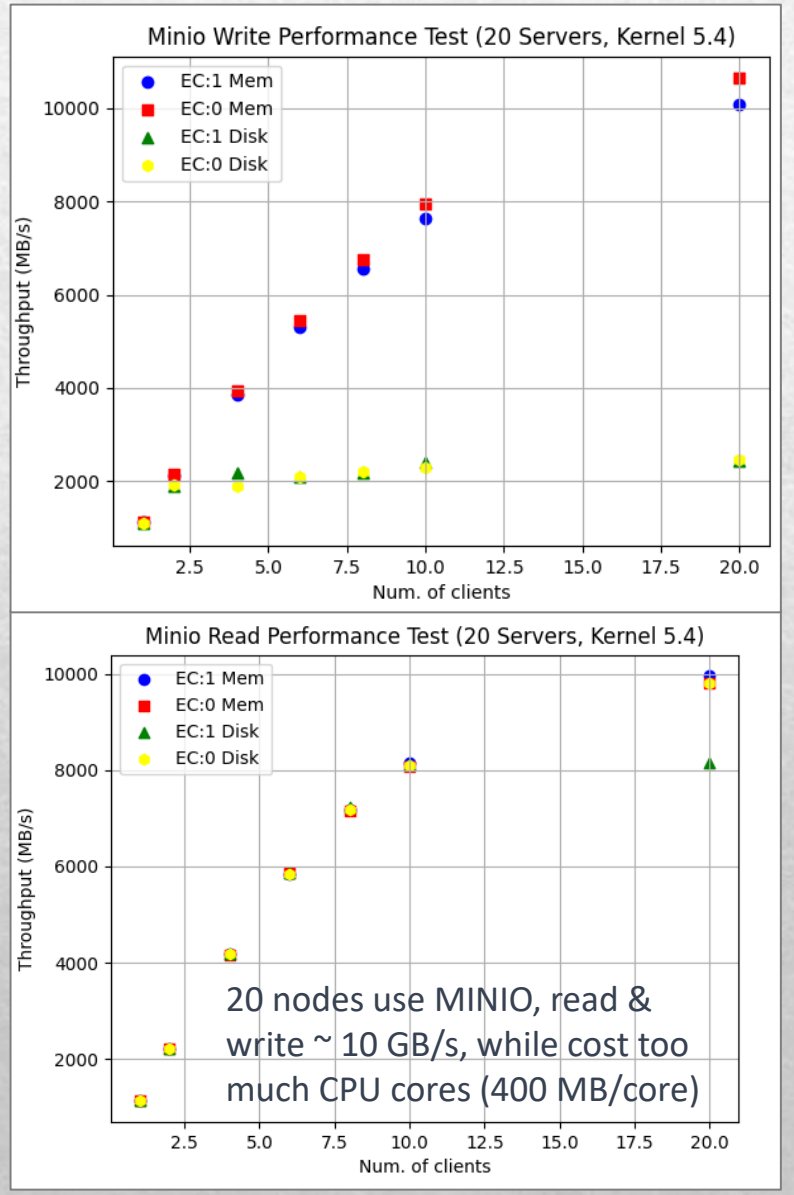
- Deploying caching pools using modified Alluxio and MINIO
- Prototype developed, applied in the LHAASO

Open-source software cannot fully meet our requirements

Next

- Employ bypass technologies to reduce CPU utilization.
- Develop high-performance distributed memory file system
- Design high reliability scheme

Self-developed high-throughput distributed memory pool on the way

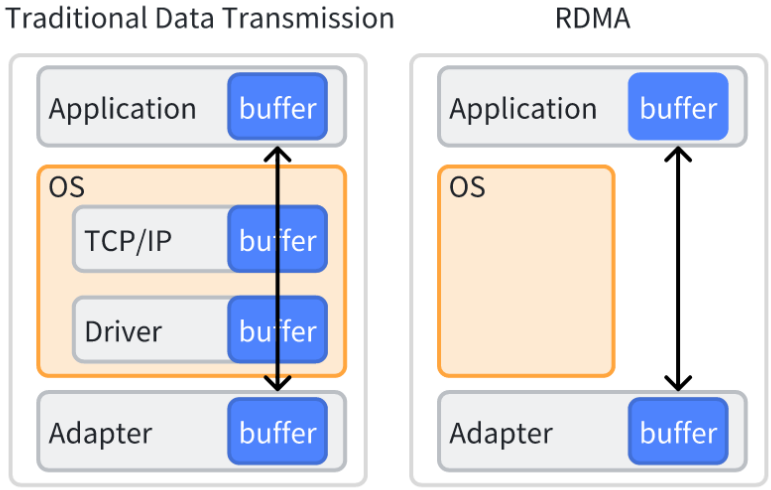


RDMA Research

RDMA (Remote Direct Memory Access)

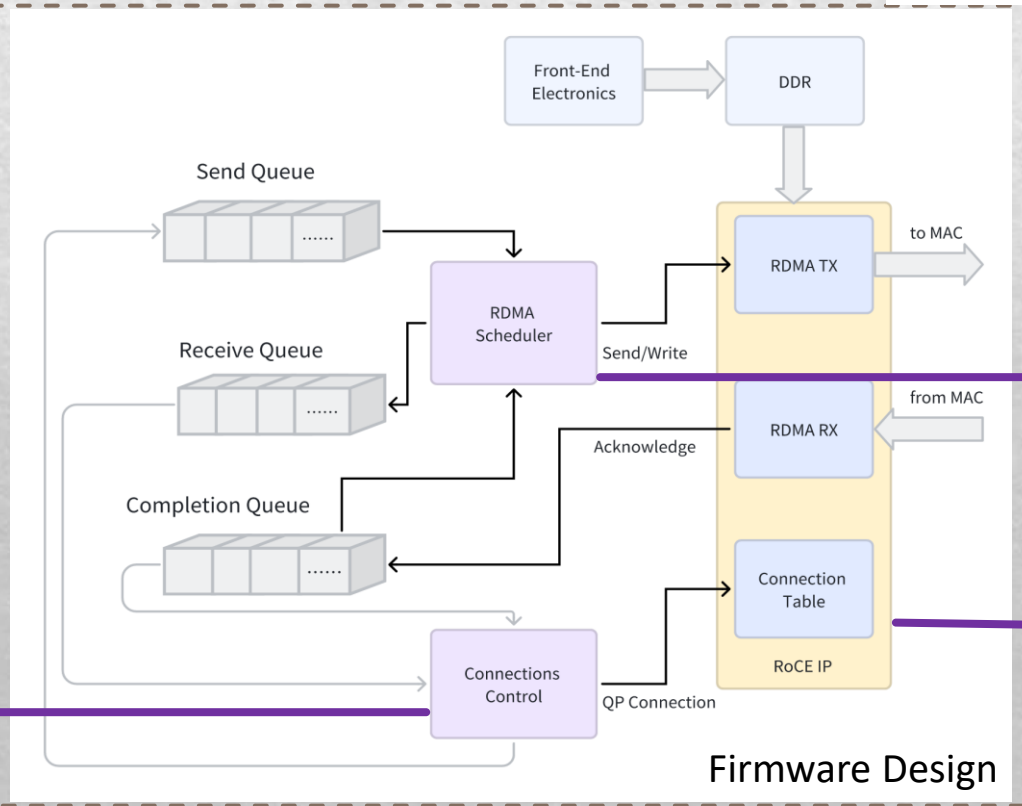
- Transfer data directly to the remote memory
- More efficient transmission with low-latency, low CPU load

Applied as DAQ readout protocol: readout module → DAQ server



Comparison of Data Transfer Processes

Implement RDMA on FPGAs



Firmware Design

Connection Control

- Maintain connections
- Support dynamic connections
- Ensure stability and reliability

RDMA Scheduler

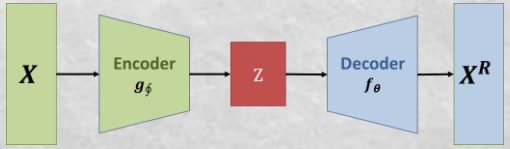
- Coordinate operations

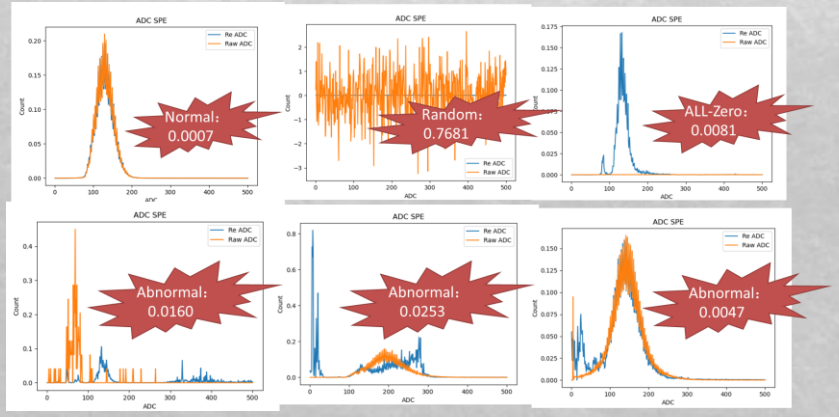
RoCE IP Core

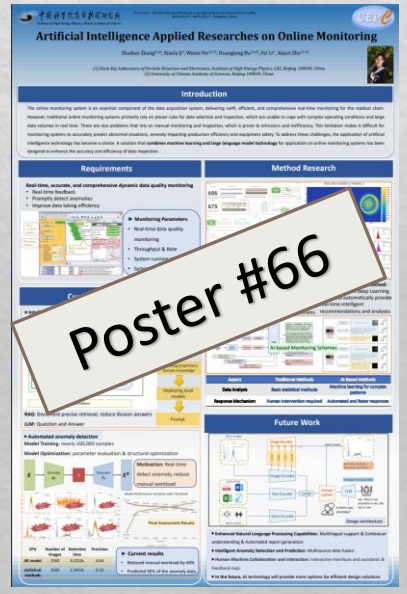
- Complete data transfer
- Exchange packets that comprise protocol

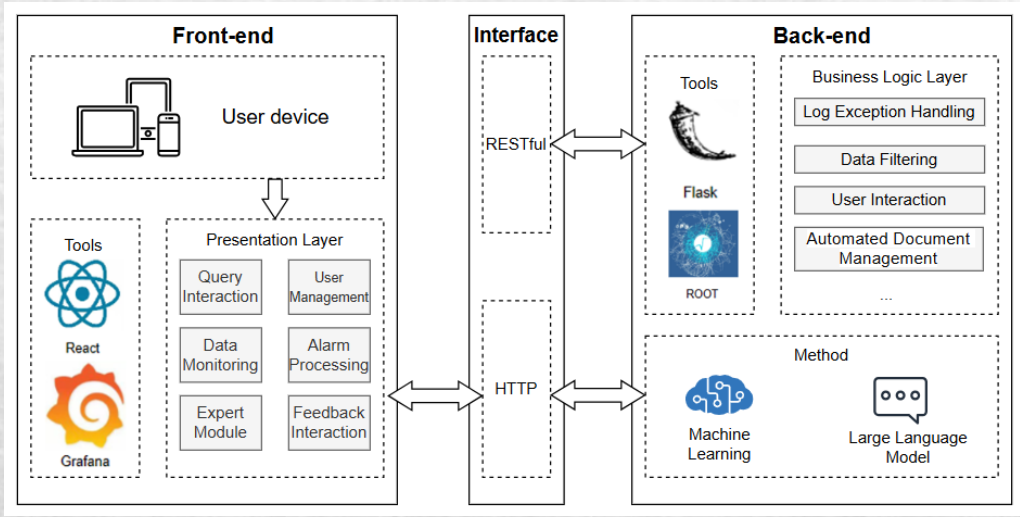
AI-based Utilities Research

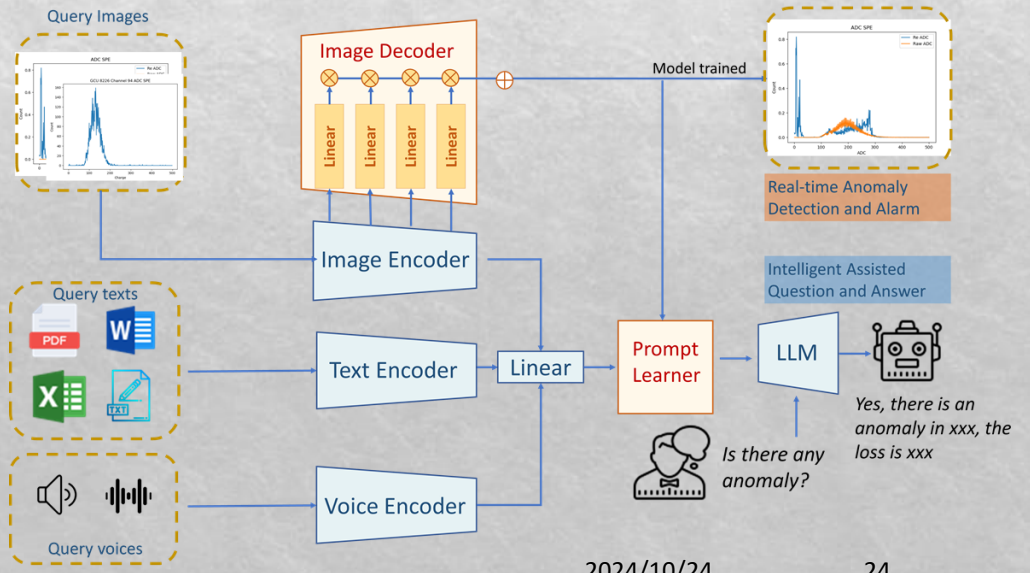
- Implement intelligent operations and improve efficiency
- Current progresses:
  - Private domain information query assistant based on LLM
  - Real-time histogram anomaly diagnosis based on ML
- Next: Cover more aspects of operations and aim for system-level applications











Summary

- We had a very preliminary technical design for CEPC DAQ
- Work in different directions is progressing
- Try some new methods and technologies
- Welcome to join us

Thanks!