JUNO DAQ: Operational Status and Recent Advances

Xiaolu Ji
Institute of High Energy Physics, CAS, China
On behalf of the JUNO Collaboration

25th IEEE Real Time Conference - La Biodola, Elba, Italy

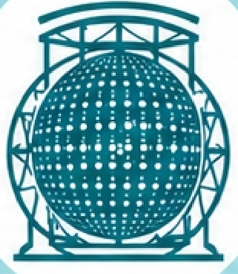
From Design Validation to Operation-driven Evolution

RT 2024: Design and Status

- Baseline architecture introduced
- DAQ performance verified with test data
- Commissioning and subsystem integration started

RT 2024 talk: JUNO DAQ Design and Status
indico.global/event/6805/contributions/58355/

RT 2026: Operational Status and Recent Advances

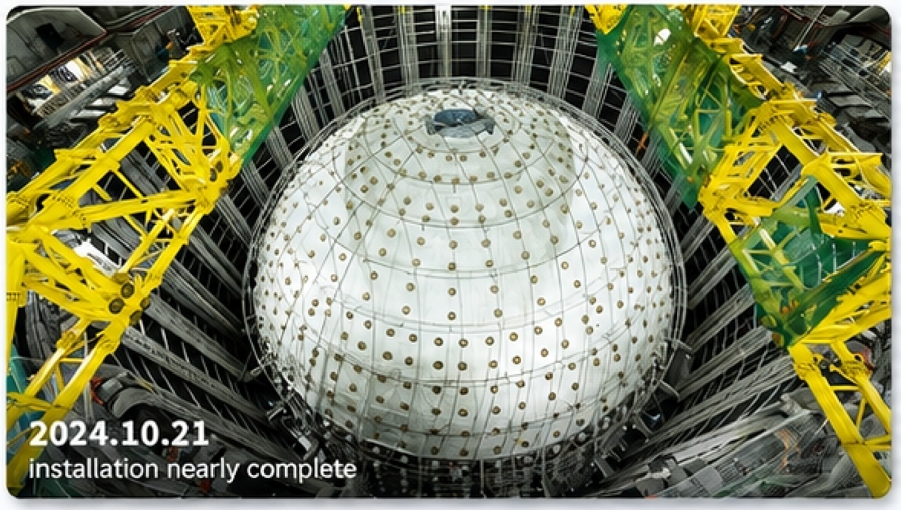


- System used in on-site detector operation
- Upgrades driven by real running conditions
- Focus on robustness, flexibility, and maintainability

This talk: brief recap of the baseline + operational status + recent advances

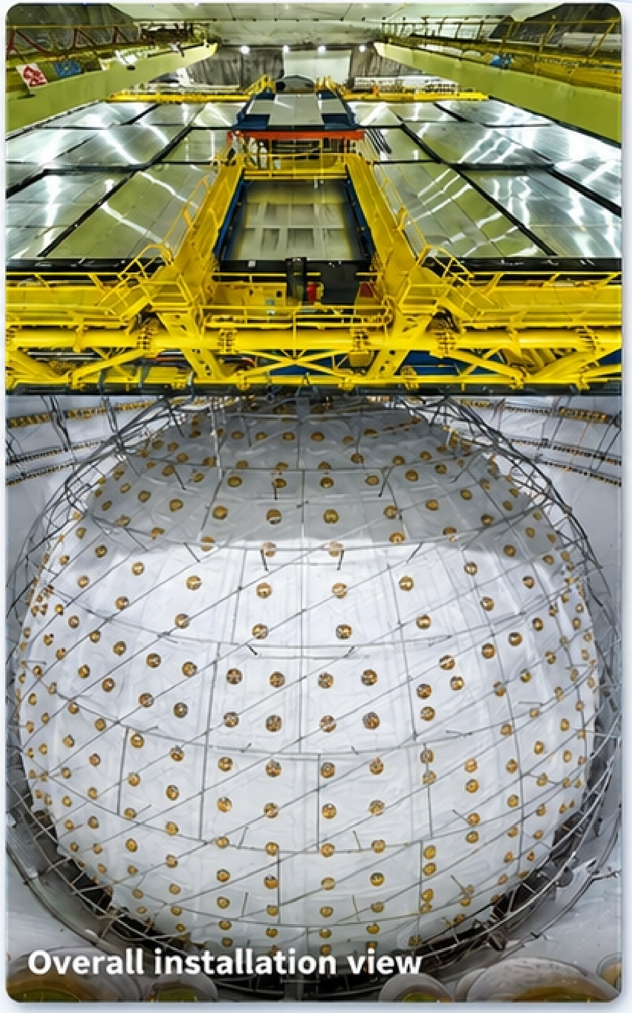
A Visual Context: JUNO Detector Installation

selected milestone views before detector operation

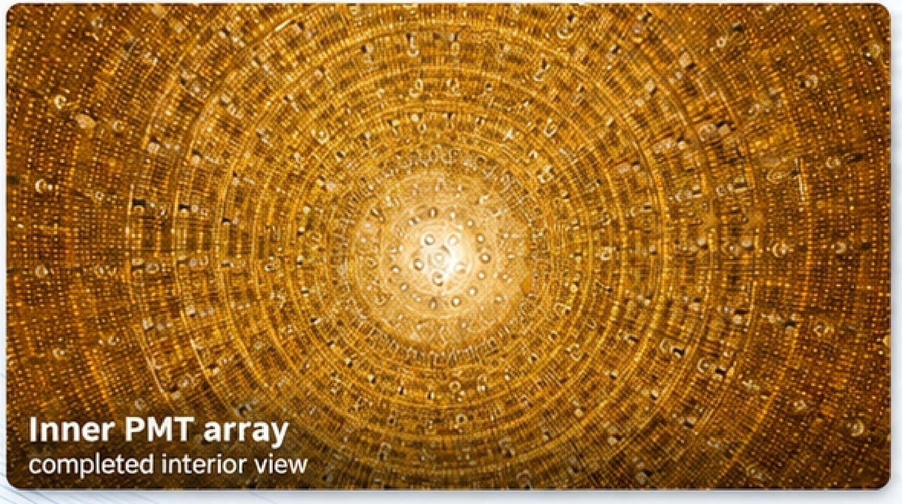


2024.10.21
installation nearly complete

2024.11.13
PMT installation nearly complete



Overall installation view



Inner PMT array
completed interior view

2024.12.18
water filling just started

JUNO in Brief and DAQ Requirements

A brief introduction to JUNO and the main requirements for its online DAQ system

JUNO in Brief

- ~700 m underground, Guangdong, China
- 20-kton liquid scintillator detector
- Construction started in 2015
- LS filling and data taking started in Feb. 2025
- Physics data taking started on Aug. 26, 2025

Detector & Readout Scale

- CD: 17,612 LPMTs + 25,600 SPMTs
- WCD: 2,400 LPMTs
- ~40 GB/s triggered waveform data
- Trigger-less T/Q data streams at a few GB/s

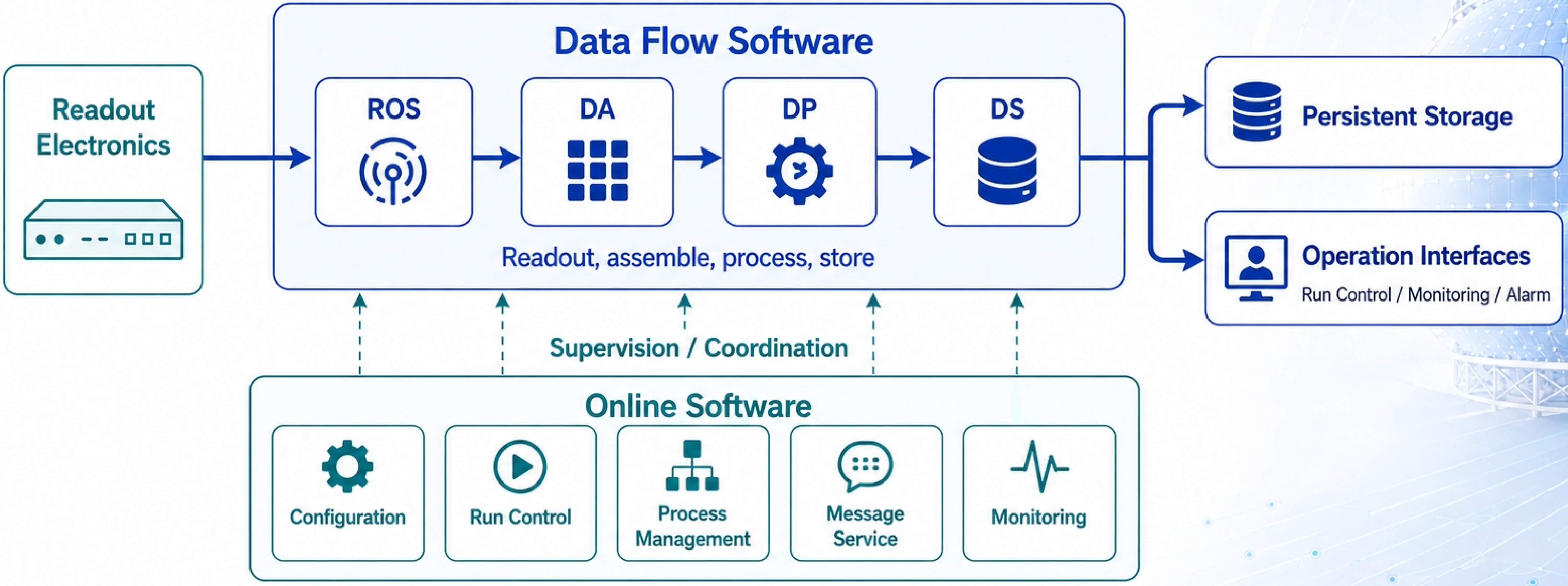
Operational Requirements

- Multiple event-building paths
- Diverse online processing tasks
- Continuous supernova monitoring
- Stable long-term operation with minimal data loss

Core requirements: high throughput + diverse online tasks + continuous reliability

JUNO: Jiangmen Underground Neutrino Observatory

Baseline Architecture of JUNO DAQ



Readout Electronics

Data Flow Software

ROS

DA

DP

DS

Readout, assemble, process, store

Persistent Storage

Operation Interfaces

Run Control / Monitoring / Alarm

Supervision / Coordination

Online Software

Configuration

Run Control

Process Management

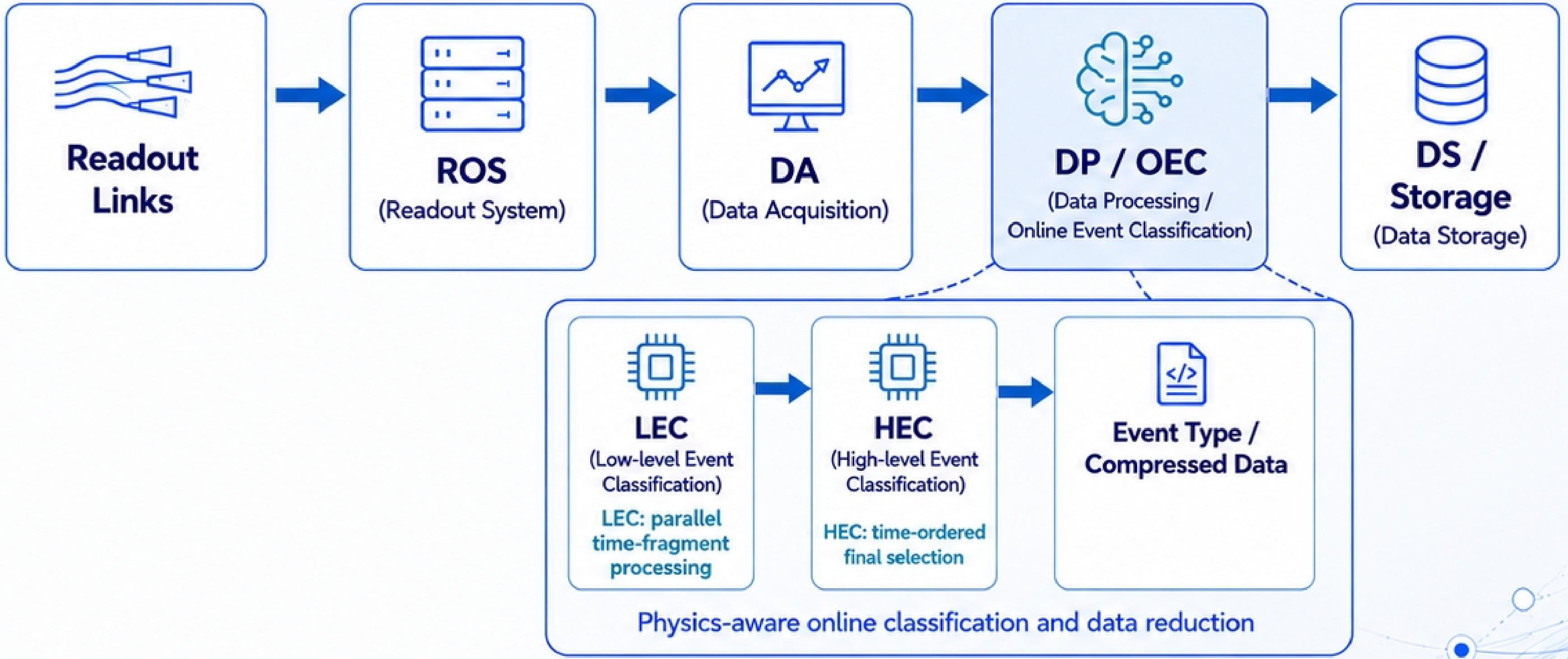
Message Service

Monitoring

Radar-based distributed framework | ZeroMQ transport | Kafka / ZooKeeper-based online services | containerized deployment

ROS: Readout System | **DA:** Data Assembly | **DP:** Data Processing | **DS:** Data Storage

Baseline Data Flow and the Role of OEC



Raw stream

~40 GB/s

↓

Storage

~60 MB/s

(with baseline OEC)

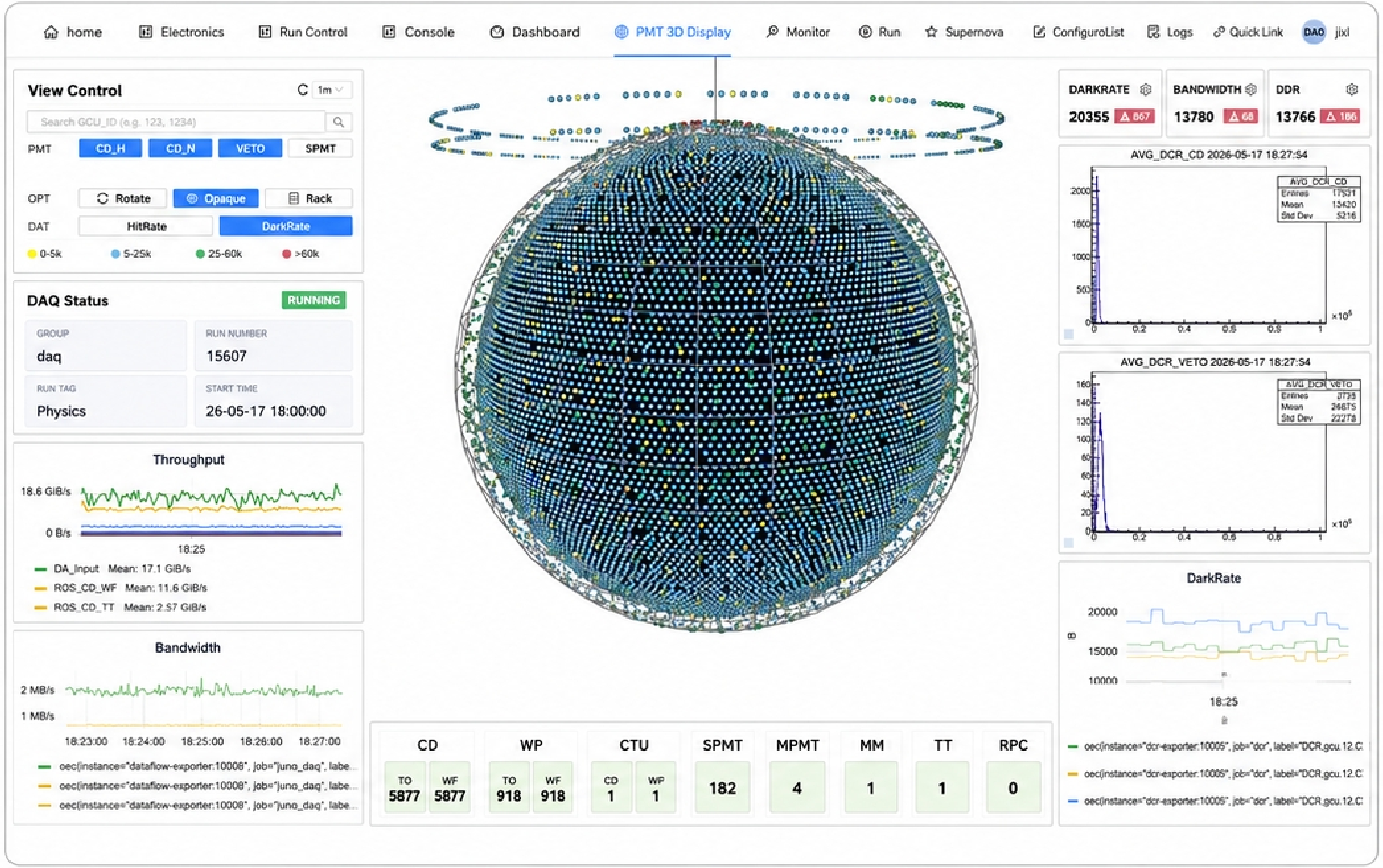
Triggered and trigger-less data are handled in a uniform, time-fragment based pipeline

- Reuse of offline reconstruction code online
- Supports flexible plugin algorithms
- Aligns online selection with offline reconstruction

OEC: Online Event Classification | **LEC:** Low-level Event Classification; **HEC:** High-level Event Classification | **ROS:** Readout System; **DA:** Data Acquisition; **DP:** Data Processing; **DS:** Data Storage

Operational Status Overview

Routine detector operation supported by online monitoring, run control, and long-term runtime tracking



DAQ Status RUNNING

GROUP daq	RUN NUMBER 15607
RUN TAG Physics	START TIME 26-05-17 18:00:00

DA_Input Mean: 17.1 GiB/s
ROS_CD_WF Mean: 11.6 GiB/s
ROS_CD_TT Mean: 2.57 GiB/s

oec{instance="dataflow-exporter:10008", job="juno_daq", labe...
oec{instance="dataflow-exporter:10008", job="juno_daq", labe...
oec{instance="dataflow-exporter:10008", job="juno_daq", labe...

DARKRATE	BANDWIDTH	DDR
20355 ⚠ 867	13780 ⚠ 68	13766 ⚠ 186

AVG_DCR_CD: Entries 17531, Mean 13420, Std Dev 5216

AVG_DCR_VETO: Entries 3738, Mean 26875, Std Dev 22278

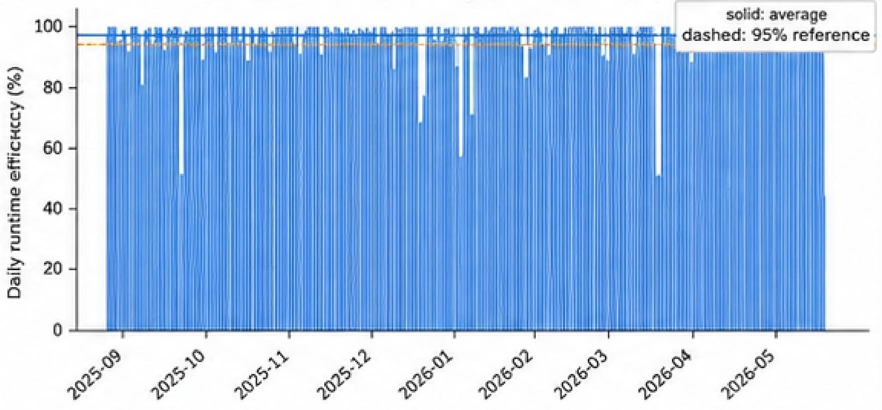
oec{instance="dcr-exporter:10005", job="dcr", label="DCR.gcu.12.C
oec{instance="dcr-exporter:10005", job="dcr", label="DCR.gcu.12.C
oec{instance="dcr-exporter:10005", job="dcr", label="DCR.gcu.12.C

CD		WP		CTU		SPMT	MPMT	MM	TT	RPC
TO	WF	TO	WF	CD	WP					
5877	5877	918	918	1	1	182	4	1	1	0

Operational Highlights

- ✓ Integrated readout from all detector systems
- ✓ Stable support for physics data taking and auto-calibration
- ✓ Comprehensive online monitoring, run control, and visualization

Daily runtime efficiency over 266 days



Average efficiency: 97.4% **Days ≥95%: 236/266** **Days ≥90%: 250/266**

Tracked period: 2025-08-26 to 2026-05-18

Current operation is supported by online monitoring, run control, and long-term runtime tracking.

Operation-driven Upgrade Roadmap

Real detector operation shaped the recent DAQ upgrade priorities

① DP Upgrade

Observation: online tasks became more diverse

Upgrade:
DAG-based flexible workflow

② RO Upgrade

Observation: burst-like trigger rates stressed fixed slicing

Upgrade:
adaptive RO time slicing

③ Mask Strategy

Observation: local channel faults could affect global dataflow

Upgrade:
dynamic and reversible masking

④ Dataflow HA

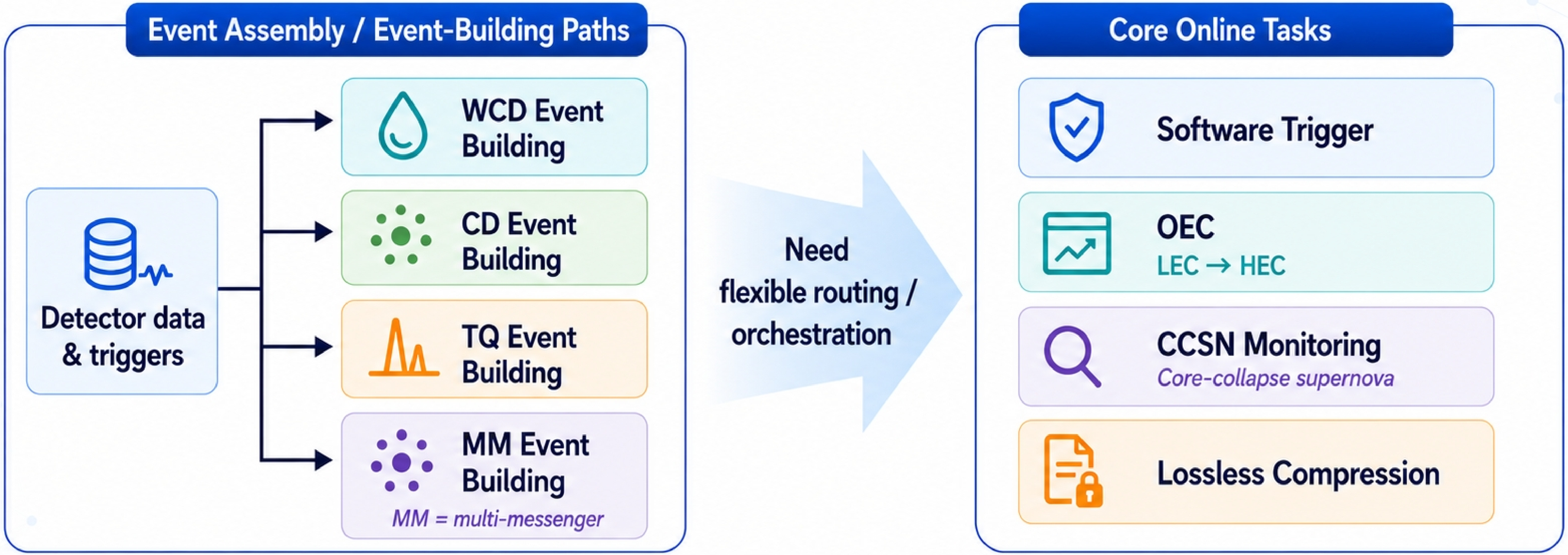
Observation: failures could interrupt continuous dataflow

Upgrade:
dataflow high-availability redesign

The following slides present the motivation, implementation, and operational effect of each upgrade.

DP Upgrade I: Why a DAG Workflow Layer is Needed

Multiple event-building paths and diverse online tasks require flexible orchestration in JUNO online processing



Event Assembly / Event-Building Paths

Detector data & triggers

- WCD Event Building
- CD Event Building
- TQ Event Building
- MM Event Building

MM = multi-messenger

Need flexible routing / orchestration

Core Online Tasks

- Software Trigger
- OEC
  LEC → HEC
- CCSN Monitoring
  Core-collapse supernova
- Lossless Compression

Multiple paths and shared data require flexible orchestration.

DP Upgrade II: DAG-Driven Online Processing Workflow

Business functions are organized as a configurable dependency graphs

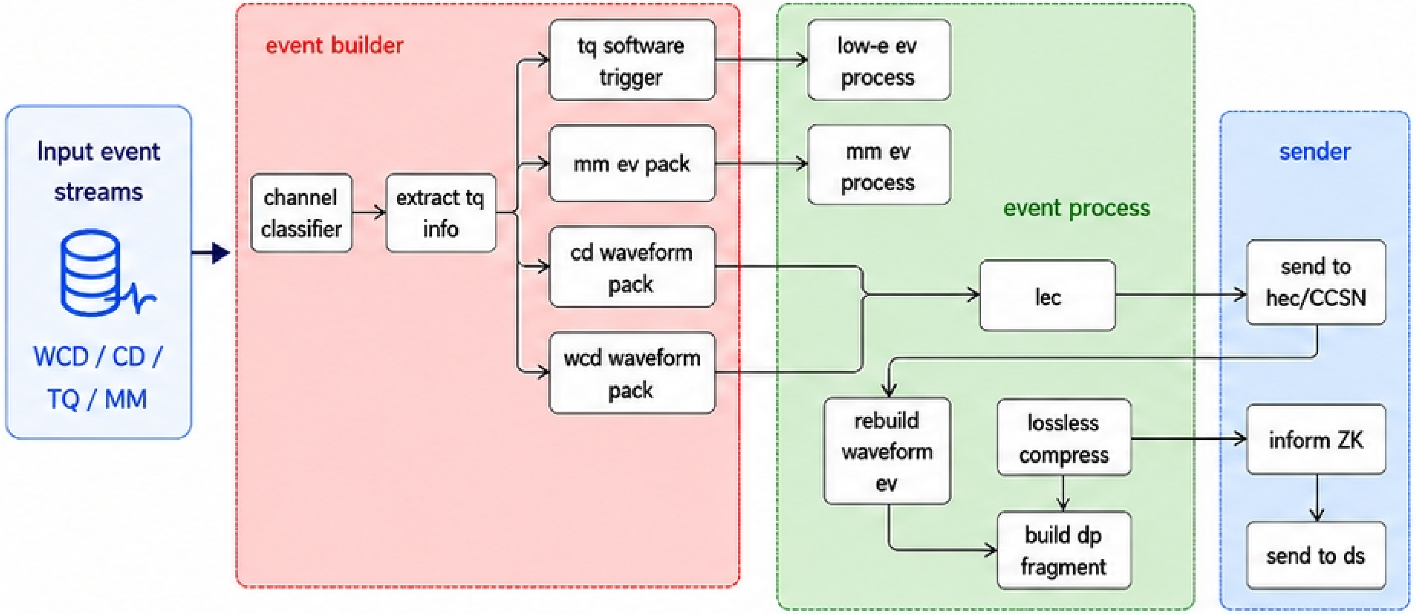
What changes with DAG

- Configured nodes and paths
- Dependency-aware scheduling
- Parallel branches

Configurable Execution Graph

Example of a configured business graph

ⓘ Core task paths supported: Software Trigger / OEC / CCSN Monitoring / Lossless Compression



Input event streams

WCD / CD / TQ / MM

event builder

channel classifier

extract tq info

tq software trigger

mm ev pack

cd waveform pack

wcd waveform pack

event process

low-e ev process

mm ev process

lec

rebuild waveform ev

lossless compress

build dp fragment

sender

send to hec/CCSN

inform ZK

send to ds

DAG = Directed Acyclic Graph; nodes represent business functions; arrows represent dependency paths.

What the upgrade enables

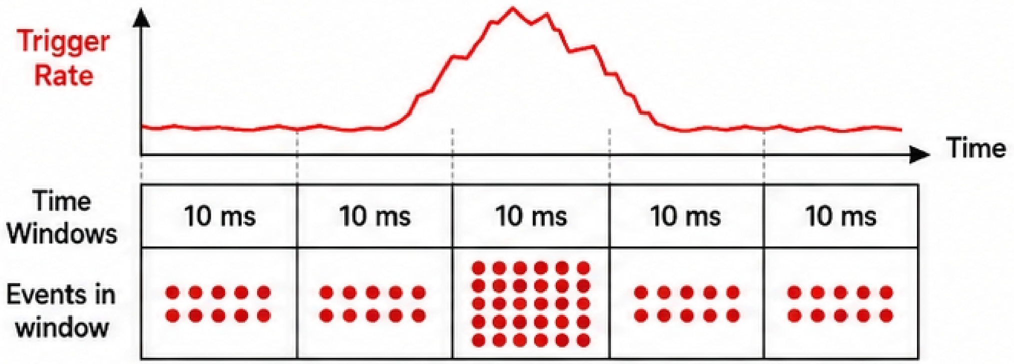
- Flexible workflow reconfiguration
- Shared modules across tasks
- Parallel execution of independent branches
- Easier integration of new online tasks

DAG makes DP workflows configurable, reusable, and easier to extend.

RO Upgrade I: From Fixed-Time to Adaptive Slicing

Load-aware slicing is designed to remain stable under burst trigger conditions

Before: Fixed-Time Slicing



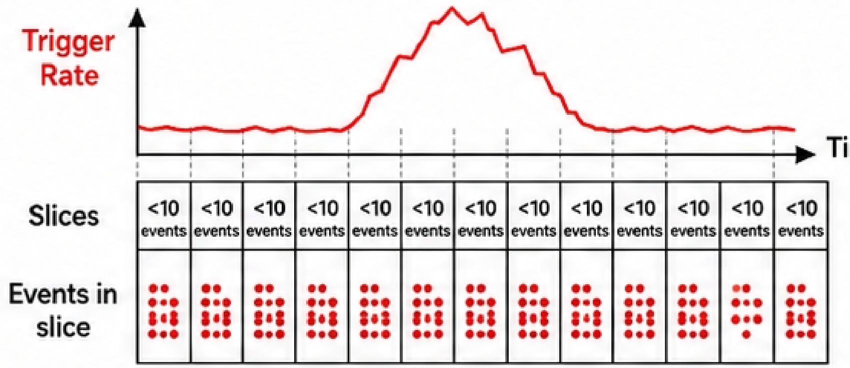
- Fixed 10 ms slice
- Burst → huge slice
- Overflow, backpressure, unstable load

Readout → Preprocess Buffer → Time Slice Builder → Processors

Problem: one burst can overload the whole chain

Upgrade

After: Adaptive Slicing



Flush when:
① events >= 10
② time >= 20 ms
③ density protection

- Bounded slice size
- Burst → more small slices
- Stable, predictable processing

Readout → Preprocess Buffer → Adaptive Slice Builder → Processors

Result: local buffers are cleared in time

Load-driven instead of time-driven | **Bounded micro-batches** | **Stable under burst**

Adaptive slicing bounds slice size and makes the readout chain more predictable under burst-like trigger conditions.

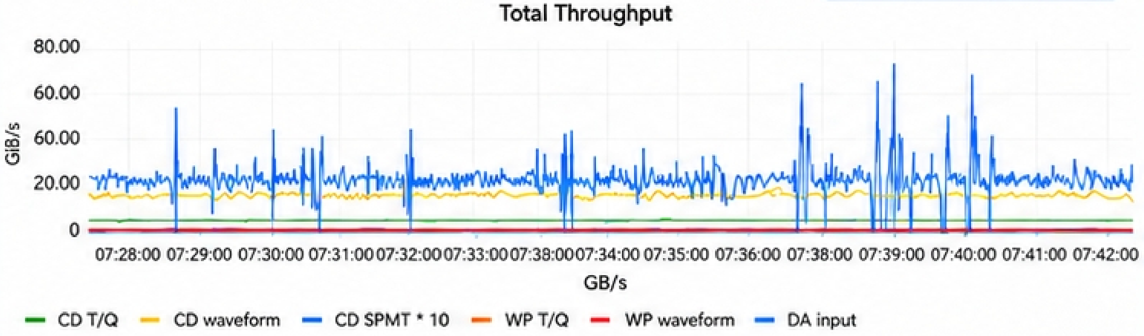
RO Upgrade II: Observed Effect in Operation

Real monitoring data show smoother throughput and stable recovery under burst-like trigger conditions

Before vs After RO Upgrade

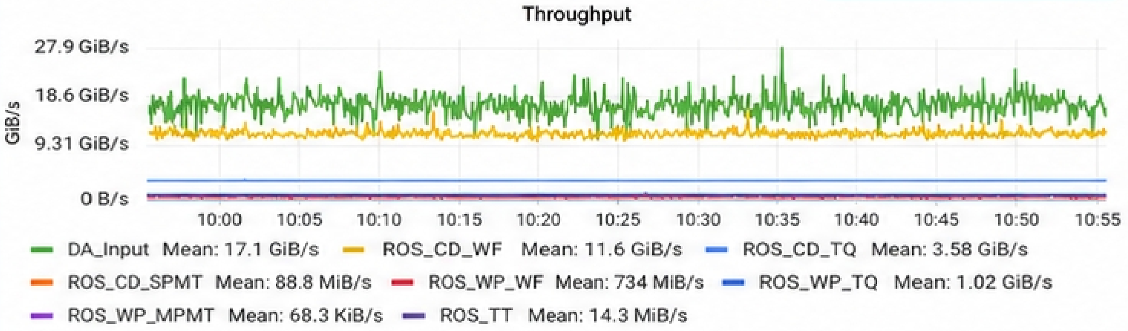
Before upgrade

Before: large throughput jitter

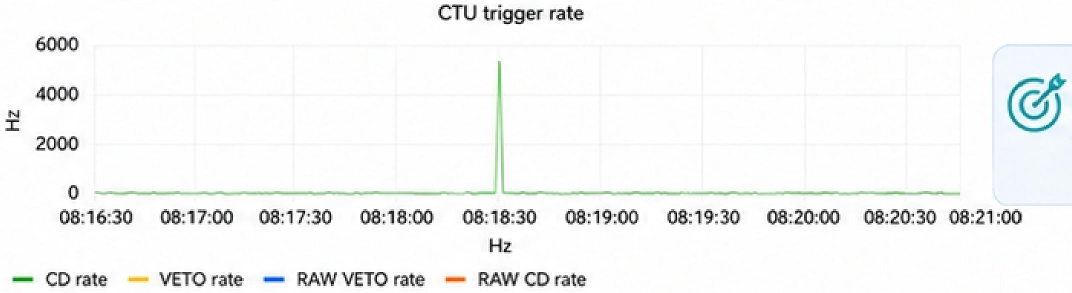


After upgrade

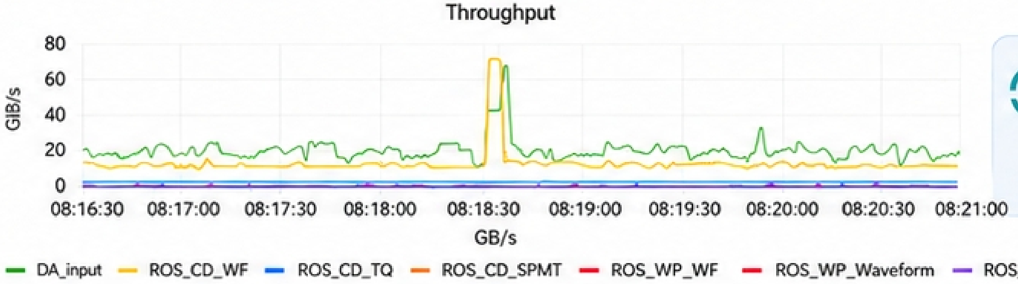
After: smoother data flow



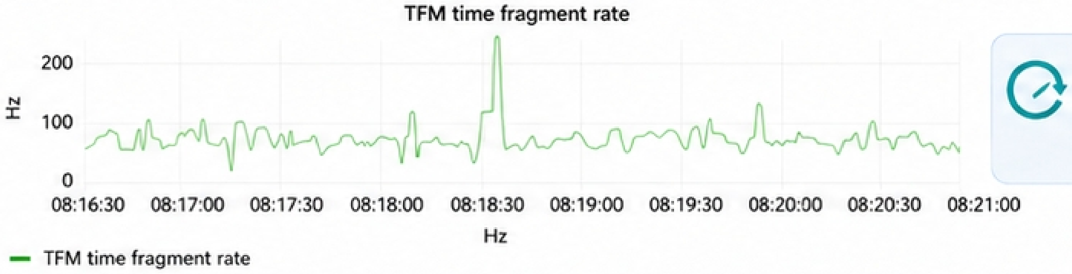
Burst-like Trigger Validation



Burst trigger rate up to ~5 kHz



Assembly throughput recovers after the burst



TFM slicing rate recovers after the burst

Adaptive RO time slicing makes the data flow more stable and more resilient to short high-rate bursts.

Mask Strategy I: Why Dynamic Channel Masking Is Needed

For continuous supernova monitoring with ultra-high channel count

Our Mission

Continuously and uninterruptedly take data (24/7) to catch rare supernova events at any time.

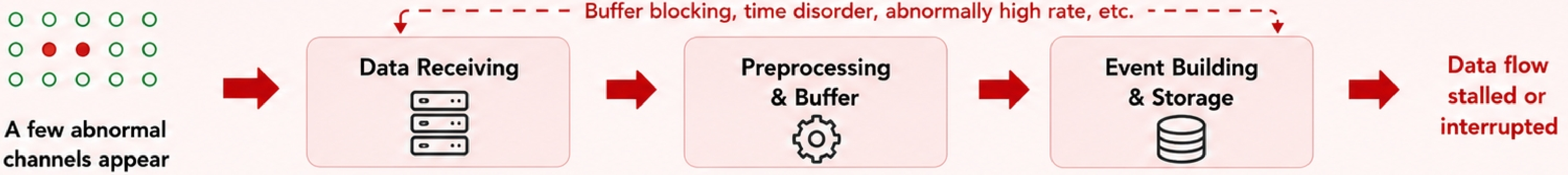
Our Scale

- ~20,000 LPMT channels
- ~25,000 SPMT channels
- ~45,000 channels in total

The Reality

A small fraction of channels will become abnormal at any time, for various reasons.

What Can Go Wrong Without Protection



Buffer blocking, time disorder, abnormally high rate, etc.

A few abnormal channels appear

Data Receiving

Preprocessing & Buffer

Event Building & Storage

Data flow stalled or interrupted

A few bad channels can block buffers, break timing, and affect the whole system.

Why Dynamic Channel Masking is Essential

Protect the Whole System
Isolate issues early to prevent cascading failures.

Keep Data Flow Continuous
Ensure uninterrupted data-taking for supernova monitoring.

Maximize Data Quality
Remove pathological channels that degrade performance.

Reduce Manual Intervention
Enable long-term autonomous and stable operation.

In ultra-high channel count systems, dynamic masking is the key to stable, continuous, and reliable data-taking.

Mask Strategy II: Workflow and Operational Validation

Dynamic masking keeps data flowing under abnormal-channel conditions

① **Monitor**
rate / timing / buffer

→

② **Detect**
find abnormal channels

→

③ **Mask**
isolate the faulty part

→

④ **Recover**
re-enable if healthy

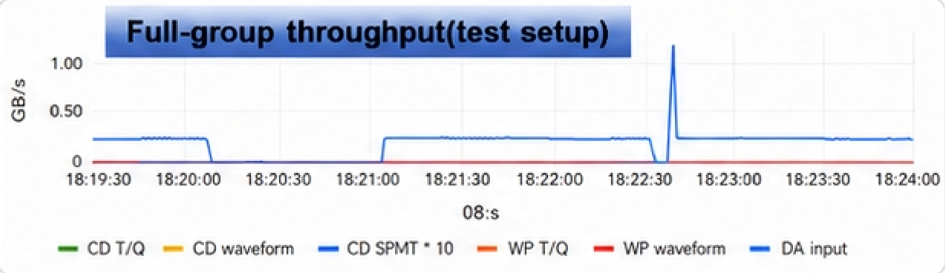
Design idea

- Fast & local
- Minimal impact
- Dynamic & reversible

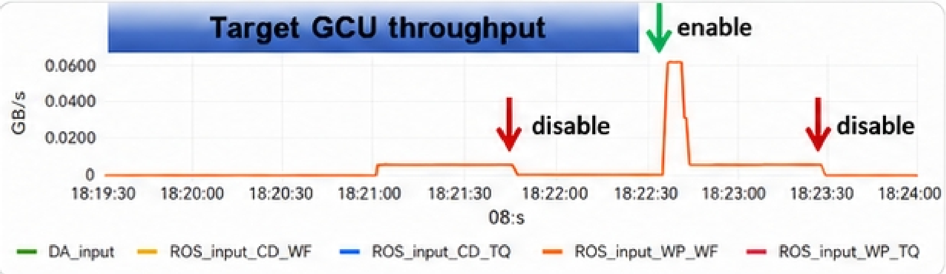
Observed Validation in Real Operation

Case 1 Single-GCU disable / recovery

Full-group throughput(test setup)

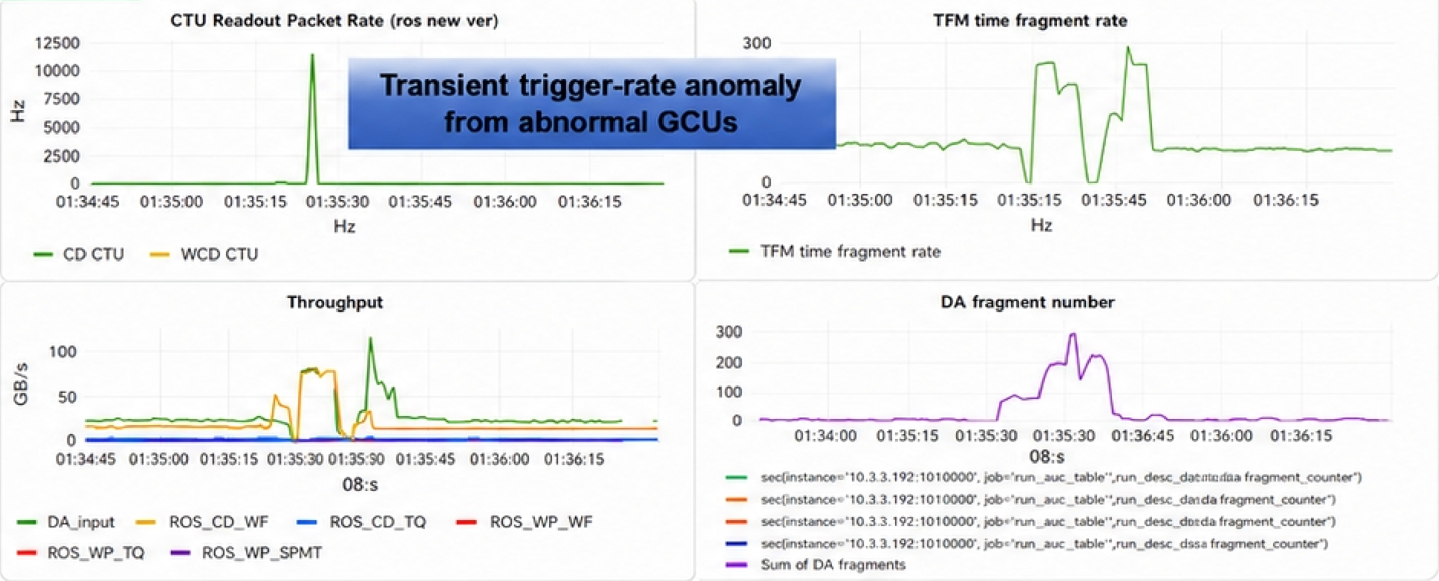


Target GCU throughput



- Other channels remain unaffected.
- Data flow recovers after buffered data are processed.

Case 2 Complex abnormal-channel scenario



Transient trigger-rate anomaly from abnormal GCUs

- Transient trigger-rate burst caused by partial GCU power loss, up to **~10 kHz**.
- Faulty GCUs isolated quickly.
- Normal bandwidth recovered within **~30 s.**
- **No DDR data loss observed.**

★ **Prevent local readout problems from propagating to the global dataflow.**

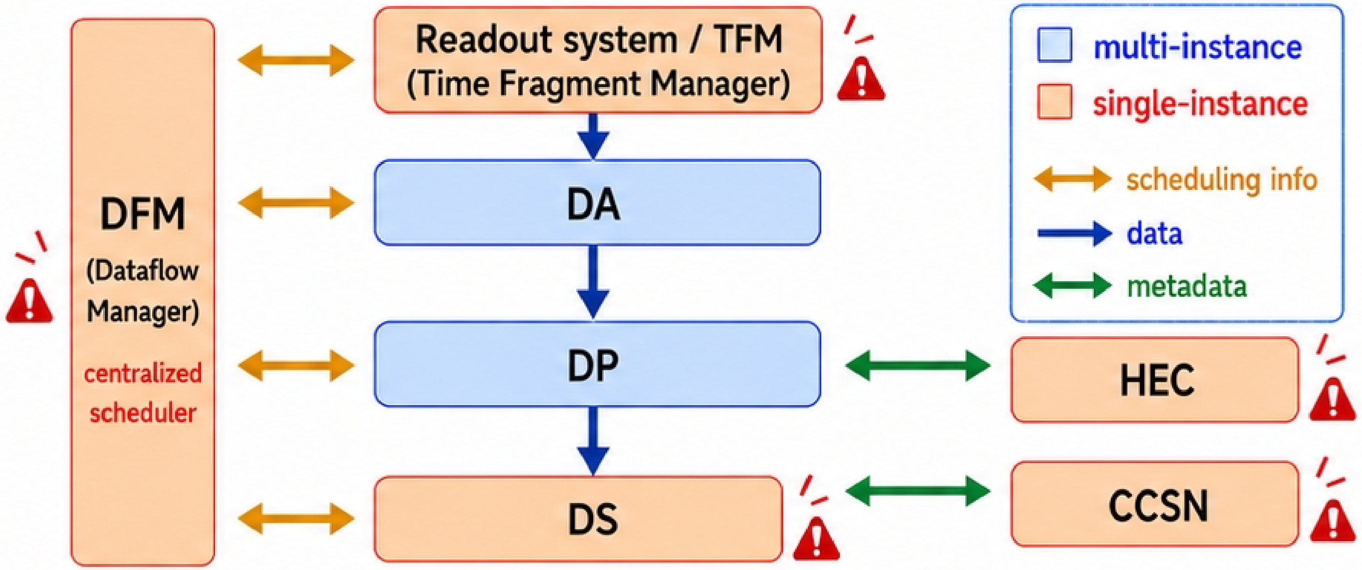
Dataflow HA Upgrade I: Why It Is Needed

Extending high availability from online services to the dataflow layer

Why it matters

1. Rare signals cannot be replayed
   supernova bursts are rare, unpredictable.
2. Large DAQ clusters face inevitable node / process failures.
3. Single-point failures may interrupt critical dataflow.

Main weakness of the original dataflow



DFM
(Dataflow Manager)
centralized scheduler

Readout system / TFM
(Time Fragment Manager)

DA

DP

DS

HEC

CCSN

🟦 multi-instance
🟧 single-instance
↔ scheduling info
→ data
↔ metadata

No dynamic service discovery

Several component-level single points

State / scheduling tied to processes

Goal: maintain continuous dataflow and reduce data-loss risk during long-term operation.

Dataflow HA Upgrade II: Redesign Strategy and Current Progress

Dynamic discovery, shared state, and distributed self-scheduling.

1. Dynamic service discovery

ZooKeeper
+
Kubernetes

- ZooKeeper tracks live instances
- Faulty services leave automatically
- Restarted services rejoin the cluster

2. State externalization and module HA

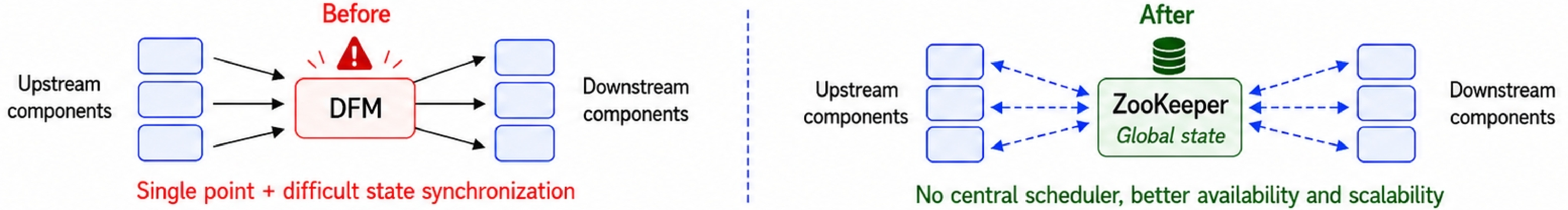
- Live state is maintained in ZooKeeper
- DA / DP move toward stateless multi-instance operation
- DS / HEC / CCSN are decoupled from fixed processes

ROS / TFM upgrades are ongoing

3. Distributed self-scheduling

- Remove centralized scheduler
- Upstream components schedule by shared state
- Recovery no longer depends on local process state

From centralized scheduling to distributed self-scheduling



Single point + difficult state synchronization

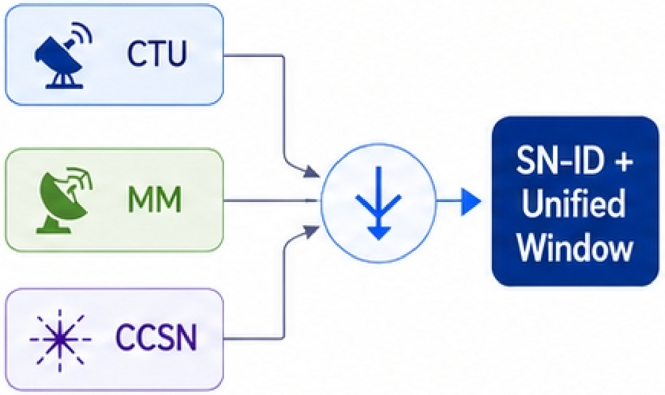
No central scheduler, better availability and scalability

Outcome: faster recovery, easier service rejoin, and stronger dataflow resilience.

Supernova Alert Handling Strategy

1 SN-ID

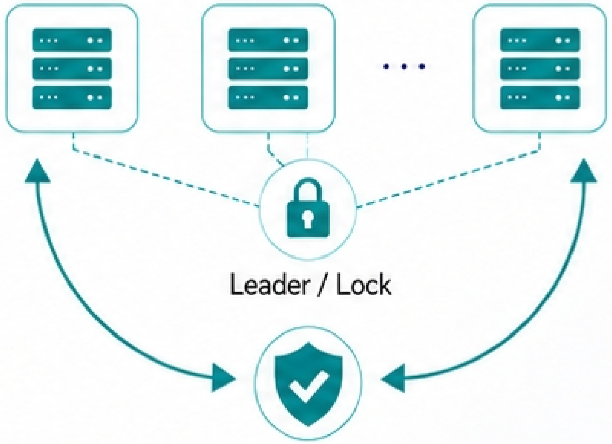
Unify multi-source alerts



✓ One event, one ID, one unified window

2 Reliable Alert Handling

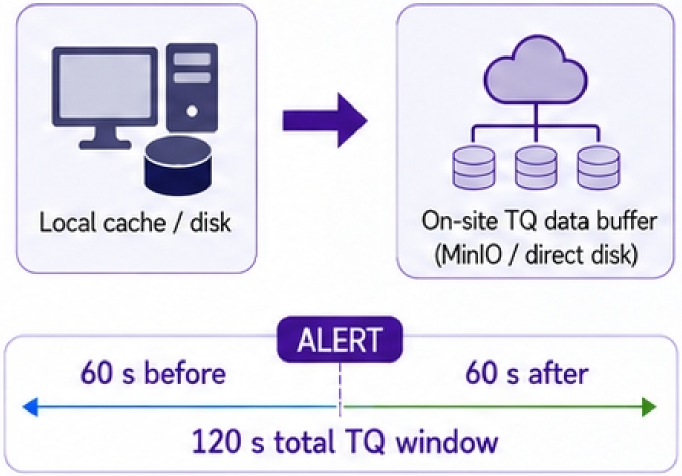
Make the alert service fault-tolerant



✓ Multi-instance, no single point of failure

3 TQ Data Buffering & Preservation

On alert: preserve 120 s TQ window



✓ 120 s TQ window saved for analysis

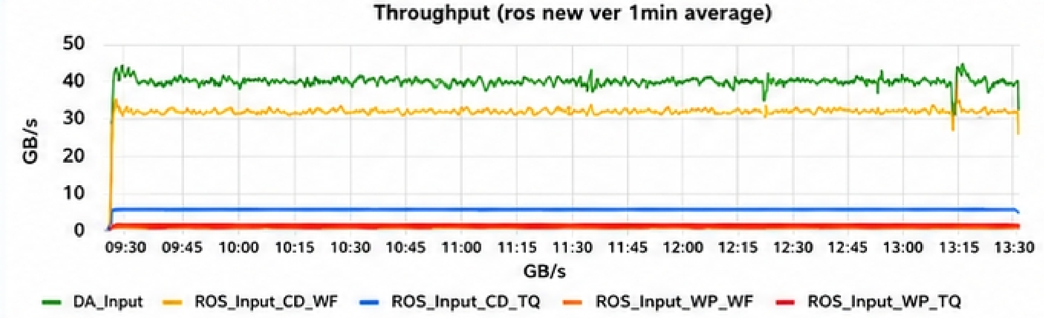
Alerts → **SN-ID + Window** → **Save 120 s TQ Data** 60 s before + 60 s after → **On-site Fast Reconstruction**

ⓘ Further details in RT2026 poster: 'Supernova processing implementation of JUNO DAQ' | https://indico.global/event/14708/contributions/149756/

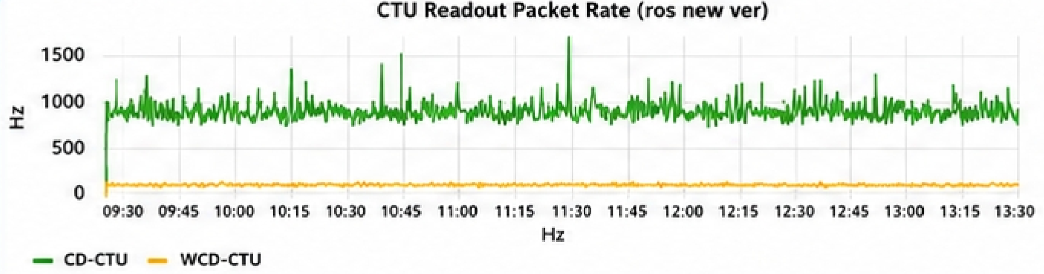
Performance Validation with Detector Readout

Real detector-readout tests confirm stable operation and operational headroom

Representative detector-readout run


Throughput (ros new ver 1min average)

✓ **Sustained throughput ~40 GB/s**

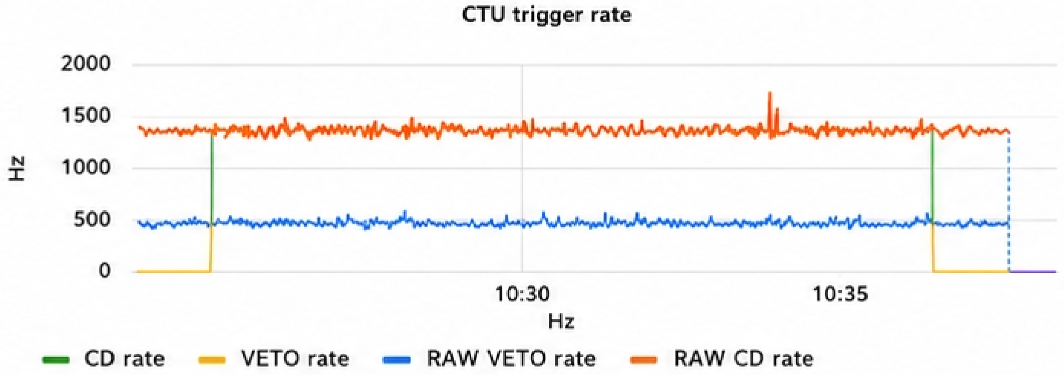

CTU Readout Packet Rate (ros new ver)

✓ **CD trigger/ readout rate ~1 kHz**

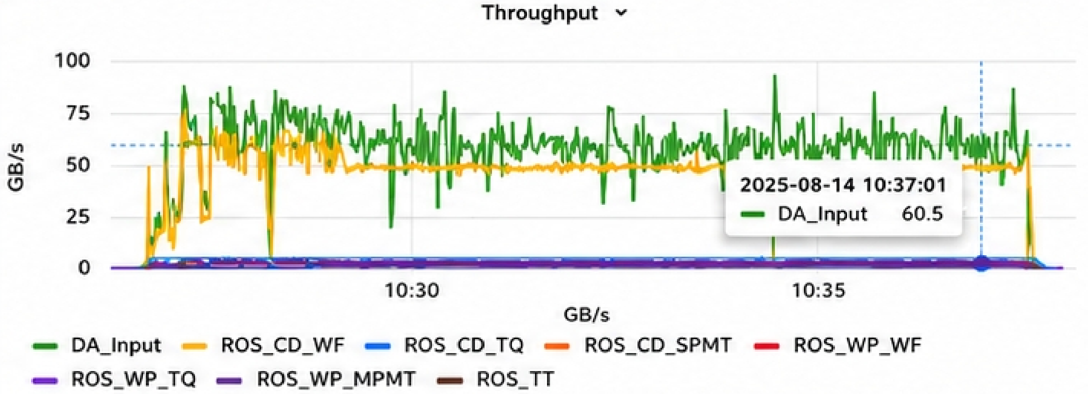

storage data rate

✓ **Storage data rate < 200 MB/s**

High-rate stress-test headroom


CTU trigger rate

✓ **Stable at 1.5 kHz**


Throughput

✓ **~60 GB/s**

~50% DAQ headroom | Typical routine operation is lower; these tests validate DAQ margin under high-rate detector-readout conditions.

Summary and Outlook

From commissioning-oriented validation to operation-driven evolution

Operational Status

- Moved from commissioning support to routine detector operation
- Supports readout, online processing, monitoring, and on-site data taking

Recent Advances

- Flexible DP workflow
- Adaptive RO time slicing
- Dynamic fault isolation
- Dataflow HA redesign

Validated on mock systems and deployed with seamless switchover.

Outlook

- Toward more automated long-term operation
- Further optimization under detector running conditions
- Continued evolution for online processing and alert handling

Key message: JUNO DAQ is evolving into an operational, maintainable, and upgradeable long-term platform.

DAQ Contributors

Contributors over the course of the project

Part of the DAQ team at JUNO

20 Contributors

Staff (7)

- Kejun Zhu
- Fei Li
- Xiaolu Ji
- Shuihan Zhang
- Minhao Gu
- Tingxuan Zeng
- Dongsheng Zhao

Ph.D. Students (6)

- Jin Li (2019)
- Zezhong Yu (2023)
- Yu Peng (2024)
- Chao Chen (2026)
- Yinhui Wu
- Xu Zhang

Master Students (7)

- Tong Zhou (2021)
- Daibin Luo (2022)
- Xiaozhao Jiang (2024)
- Wenxi Pei (2025)
- Xiangyi Mu
- Yuan Shi
- Hao Wang

** Years in parentheses indicate graduation year*

Acknowledgement: JUNO DAQ progress relies on long-term contributions from staff and students across design, development, commissioning, and operation.