Data Acquisition System for the CSR External-Target Experiment

Authors: Sun, Mr. Zhengyang, Yang, Dr. Junfeng, Zeng, Mr. Jinrui, Sun, Mr. Ke, Li, Miss Lun, Bi, Miss Yulin, Huang, Mr. Yiting, Li, Mr. Yi, Xiong, Mr. Zilong, Yang, Dr. Junfeng

Date: 2026-01-30T16:59:55+00:00

Abstract

The CSR External-target Experiment (CEE) represents China's first large-scale nuclear physics facility operating in the GeV energy regime. To address the challenges of high data throughput and complex system integration, this paper presents the design and implementation of the CEE Data Acquisition (DAQ) system. Built upon the D-Matrix platform, the system employs a firmware-software co-designed stream processing architecture. By adopting a modular design strategy with standardized interfaces, the system achieves exceptional flexibility and scalability, unifying the design methodology across hardware and software domains. The hardware architecture features a hierarchical aggregation topology utilizing custom CROB-PXI and CROB-PCIe boards, while the software stack supports dynamic global control and high-performance data handling. System-level performance evaluations demonstrate an aggregate event building throughput of 11.74 GB/s (at 27202 Hz) and a cluster storage speed of 6.72 GB/s. These results significantly surpass the design requirements, ensuring reliable operation for the CEE experiment. Furthermore, the successful deployment of this system validates the D-Matrix architecture, establishing a reference framework next-generation experiments such as HFRS and STCF.

Full Text

Preamble

Data Acquisition System for the CSR External-Target Experiment∗
Zheng-Yang Sun,1 Jun-Feng Yang,1, † Jin-Rui Zeng,1 Ke Sun,1 Lun Li,1 Yu-Lin Bi,1 Yi-Ting Huang,1 Yi Li,1 and Zi-Long Xiong1
1Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China

The CSR External-target Experiment (CEE) represents China's first large-scale nuclear physics facility operating in the GeV energy regime. To address the challenges of high data throughput and complex system integration, this paper presents the design and implementation of the CEE Data Acquisition (DAQ) system. Built upon the D-Matrix platform, the system employs a firmware-software co-designed stream processing architecture. By adopting a modular design strategy with standardized interfaces, the system achieves exceptional flexibility and scalability, unifying the design methodology across hardware and software domains. The hardware architecture features a hierarchical aggregation topology utilizing custom CROB-PXI and CROB-PCIe boards, while the software stack supports dynamic global control and high-performance data handling. System-level performance evaluations demonstrate an aggregate event building throughput of 11.74 GB/s (at 27 202 Hz) and a cluster storage speed of 6.72 GB/s. These results significantly surpass the design requirements, ensuring reliable operation for the CEE experiment. Furthermore, the successful deployment of this system validates the D-Matrix architecture, establishing a reference framework for next-generation experiments such as HFRS and STCF.

Keywords: Data acquisition, CEE, heavy-ion experiments, stream processing, distributed system design.

Introduction

The CEE [1] is the first large-scale nuclear physics experimental facility in China operating in the GeV energy region. The CEE is stationed at the Heavy Ion Research Facility at Lanzhou Cooling Storage Ring (HIRFL-CSR) [2–4], a facility capable of delivering a variety of heavy ions of elements from hydrogen to uranium, with beam energies spanning from hundreds of MeV/u to several GeV. This capability allows the CEE to characterize low-temperature, high-density baryonic matter, making it a key platform for studying the phase diagram of Quantum Chromodynamics (QCD) [5]. Table 1 shows the main technical performance of the CEE.

As shown in Fig. 1 [FIGURE:1], the CEE spectrometer comprises a dipole magnet and a suite of detector systems, including a beam monitor [6–10], a start time detector (T0) [11–13], a time projection chamber (TPC) [14–16], an inner time-of-flight detector (iTOF) [17–20], a multi-wire drift chamber (MWDC) [21–25], an external time-of-flight detector (eTOF) [17, 26–29], and a zero-degree calorimeter (ZDC) [30–32]. Additionally, the CEE incorporates essential common subsystems, including the DAQ [33–36], trigger [37], clock [38–40], and Slow Control System (SCS) [41].

Table 1. Design performance of CEE

Parameter	Value
Maximum beam energy for proton	0.5 GeV/u
Maximum beam energy for uranium	10 kHz

Parameter	Value
Maximum event rate	$> 50\%$
Acceptance	20,000
Total channel number	

∗ Supported by the National Natural Science Foundation of China (NSFC) (No. 11927901 and No. 12275268)
† Corresponding author, Jun-Feng Yang, University of Science and Technology of China, 96 Jinzhai Road, Baohe District, Hefei, Anhui 230026, China. Tel: +86-138-5605-2505, E-mail: yangjf@ustc.edu.cn

In large-scale physics experiments, the DAQ system is responsible for the rapid and complete acquisition of data from readout electronics, real-time processing tasks such as event building, data storage, as well as system configuration and status monitoring. The CEE DAQ system is implemented based on D-Matrix [33–36, 42], a generic firmware-software co-designed stream processing platform, involving in-depth research into distributed architecture design, protocol design, stream processing modules, and performance optimization.

II. Requirements

This section discusses the requirements imposed on the DAQ system, which are categorized into functional requirements and performance specifications.

A. Functional Requirements

The functional requirements for the DAQ system are summarized as follows:

1. **Data Acquisition and Processing**: The CEE operates in a triggered mode, requiring the DAQ system to aggregate data from all detector channels and execute event building by matching trigger IDs, ultimately recording complete event files to storage.
2. **Run Control and Management**: The DAQ system is responsible for the overall control and management of all experimental components. This encompasses access to and management of detector Front-End Electronics (FEEs), the control of run parameters and operational phases, as well as synchronization management in coordination with the trigger and clock systems.
3. **System Status Monitoring**: The DAQ system is required to monitor the overall operational status of the experiment, which includes:
    - Global trigger and data rates, as well as traffic information for individual channels;
    - Operating conditions of electronic components, such as temperature, humidity, and electrical currents;

- Transmission of operational status data to the slow control system for display and archiving;
- Sampling of built event files and their transmission to the scientific application system for online analysis and visualization.

B. Performance Specifications

The performance specifications of the DAQ system are listed in Table 2 . Specifically, the real-time event building capacity is derived from the maximum design event rate of the spectrometer presented in Table 1. The data throughput specification is derived from the estimated peak data rates of the individual detector systems, as presented in Table 3 .

Table 2 . Key design specifications of the CEE DAQ system

Parameter	Specification
Real-time data throughput	$>$ 5 GB/s
Real-time event building capacity	$>$ 10 000 events/s
Data storage rate	900 MB/s

In terms of storage specifications, the CEE experiment utilizes a beam with slow extraction, resulting in a periodic time structure (as shown in Fig. 2 [FIGURE:2]). The beam period is approximately 25 s, containing a spill length of about 5 s. This corresponds to a beam duty cycle of roughly 20%. Consequently, the data acquisition system does not require continuous storage at the peak data rate. Instead, the storage bandwidth is designed to match the average data rate, which is 20% of the peak rate, resulting in a requirement of approximately 900 MB/s.

Table 3 . Estimated peak data rates for the CEE detector systems

Detector System	Total Channels	Peak Rate (MB/s)
iTOF & eTOF	$<$ 4200	4500

a The 216 FEEs consist of 132 for iTOF and 84 for eTOF. As they share identical hardware designs and readout logic, the DAQ system handles them without differentiation.

III. Architecture

This section presents the overall architecture and technical framework of the CEE DAQ system.

A. Overall Hardware Architecture

The CEE DAQ system adopts a hierarchical aggregation architecture, with its overall hardware structure shown in Fig. 3 [FIGURE:3]. This structure can be broadly categorized into two primary segments: the FPGA-based boards and the back-end server cluster.

1. FPGA Boards The FPGA-based boards comprise two types of Common ReadOut Boards (CROBs): the CROB-PXI (Common ReadOut Board on PXI) and the CROB-PCIe (Common ReadOut Board on PCIe).

The CROB-PXI is installed in a PXI chassis within the experimental hall. It establishes bidirectional point-to-point links with multiple FEEs via optical fibers and is responsible for the scientific data readout, the aggregation of status, urgent messages, and command feedback, as well as command fan-out.

The CROB-PCIe is installed in the readout server within the near-end server room. It establishes bidirectional point-to-point links with multiple CROB-PXI boards via optical fibers, performing secondary aggregation of various types of information and transmitting them to the server cluster via the PCIe interface.

2. Servers The DAQ server cluster comprises four types of servers: Data Access Servers (DAS), Event Building Servers (EBS), the Online Control Server (OCS), and the Remote Online Control Server (ROCS).

Equipped with CROB-PCIe boards, the DAS serves as the bridge for data readout and hardware-software communication. It subsequently distributes data to various EBS nodes via a fully connected data switching network.

The EBS performs the final full event building and writes the resulting event files to a Network Attached Storage (NAS) cluster.

The OCS is responsible for system-wide run control and monitoring services. It acts as a relay node for non-experimental information—such as commands, status updates, and urgent messages—and forwards status data to the SCS for monitoring.

Unlike the aforementioned servers, the ROCS is stationed in the main control room rather than the near-end server room. Serving as the Human-Machine Interface (HMI) for the DAQ system, it enables operators to control and monitor the entire system by interacting with the OCS via a web-based interface.

B. D-Matrix Platform

The CEE DAQ is implemented based on the D-Matrix platform [42]. D-Matrix is a general-purpose, heterogeneous, and distributed stream processing platform characterized by the following key features: - **Stream processing paradigm**: By abstracting various information payloads into pure streams, the system significantly reduces processing coupling. These streams remain logically inde-

Figure 4

Figure 1: Figure 4

pendent while being multiplexed over a shared physical link. - **Standardized data framing**: By defining standard data frames through fine-grained protocol standardization, the platform facilitates generic processing implementation within FPGAs. - **Unified design methodology**: The platform employs generic hardware encapsulation definitions to unify hardware and software design approaches, effectively realizing hardware-software collaborative processing. - **Modular construction**: The processing system is built using a library of standardized, cascadable stream processing modules, ensuring high system flexibility and scalability.

C. Protocol Design

To accommodate distinct information types, the CEE DAQ defines the following specific information streams: - **Data stream**: Carries scientific data transmitted by the detectors. - **Command stream**: Dedicated to system control and configuration instructions. - **Command feedback stream**: Contains return values or responses from read commands. - **Status stream**: Transmits various monitoring metrics, such as voltages, currents, and data traffic. - **Urgent message stream**: Conveys messages requiring immediate action, such as alerts for voltage or current threshold violations.

All distinct information streams are processed using a unified processing paradigm and adhere to a standardized frame format definition, as illustrated in Fig. 4

, where individual streams are differentiated by the Stream ID field (part of the Stream Frame Head) within the frame header. The frame structure comprises three distinct segments: header, data block, and trailer sections. The data block functions as a subordinate data structure within the frame, enhancing the organizational flexibility of the payload. In the payload segment, this data model provides four data organization schemes to address heterogeneous output formats across subsystems. The detailed specifications of this data frame container are provided in [34].

D. Modular Design

The CEE DAQ employs a modular design strategy to construct a firmware-software co-designed stream processing system. Processing chains for individual streams are established by cascading specific, configurable firmware and software modules. To facilitate this, a suite of standardized stream processing modules has been developed. These stream processing modules can be broadly categorized into flow control modules and application processing modules.

To enable arbitrary cascading capabilities, a unified interface model is required. Firmware modules are interconnected using either the standard AXI4-Stream interface or the custom Standard D-Matrix FPGA (SDMF) interface, whereas software modules utilize the custom Standard D-Matrix Software (SDMS) interface. The detailed designs of these modules and interfaces are presented in subsequent sections.

E. Hardware Abstraction and Interface Encapsulation

The CEE DAQ adopts a layered design philosophy by introducing the concept of a Board Support Package (BSP). This approach effectively isolates hardware details from upper-layer data processing, thereby achieving platform independence for the application layer. The BSP corresponds one-to-one with the specific hardware board design. Consequently, the BSP design remains identical even if the boards host different application layer logic.

As illustrated in Fig. 5 [FIGURE:5], the BSP primarily encapsulates two categories of content. Regarding hardware resources, it encompasses the control logic for on-board chips and physical interfaces. In terms of transport interfaces, it encapsulates the underlying transport protocols and the Multiple Point-to-Point (MPP) model. Leveraging this low-level encapsulation, the application layer logic connects to the BSP exclusively via the AXI4-Stream interface to transmit specific streams.

The MPP model, illustrated in Fig. 6 [FIGURE:6], serves as the inter-node transmission interface model within the D-Matrix platform. Its core characteristic lies in the encapsulation of multiple independent streams over a single physical link, where each stream is associated with its own independent source and destination application layer modules. The MPP module incorporates critical features such as priority scheduling, retransmission, and backpressure.

IV. Hardware and Firmware Design

This section presents the FPGA firmware logic design and the hardware design of the CEE DAQ system.

A. Firmware Module Interfaces

Defining a standardized module interface is a prerequisite for enabling module interconnection. To enhance processing flexibility, the CEE DAQ adopts two distinct interfaces for firmware modules depending on their function and location: the AXI4-Stream interface and the SDMF interface.

1. AXI4-Stream Interface As a subset of the AMBA AXI4 protocol [43], AXI4-Stream [44] is optimized for high-speed data streaming by eliminating address channels to enable high-efficiency burst transmission. Due to its status

as a standard interface in the Xilinx Vivado ecosystem with extensive IP support, the CEE DAQ adopts AXI4-Stream as a fundamental firmware interface. Specifically, it is utilized for interconnecting modules that require raw serial data transmission without frame parsing, such as in transport layers or link merging scenarios.

In the CEE DAQ firmware architecture, module interconnection relies on a specific subset of AXI4-Stream signals (Fig. 7 [FIGURE:7]). Flow control and backpressure are managed via the standard TVALID/TREADY handshake, while the payload signals TDATA (data), TKEEP (byte qualifier), and TLAST (packet boundary) handle the effective information transfer.

2. SDMF Interface For modules within the FPGA application layer that require data parsing and processing, the custom SDMF interface defined within D-Matrix is adopted. The SDMF interface serves as an extended encapsulation of the AXI4-Stream interface, sharing identical basic operating logic and handshake mechanisms.

As illustrated in the timing diagram in Fig. 8 [FIGURE:8], the SDMF interface introduces two key modifications: - It incorporates support for the data block substructure to facilitate fine-grained, generic data definitions, thereby extending the capability of generic processing implementations within FPGAs. - It exposes spatiotemporal attributes alongside the data payload during transmission, enabling the receiver module to simultaneously acquire both data and spatiotemporal information. This mechanism significantly enhances pipeline processing efficiency, which is critical for the overall performance of the stream processing system.

B. Standard Firmware Stream Processing Modules

Based on the modular architecture, a library of standard firmware modules has been implemented and categorized into two layers: the Transport Layer for data flow management and protocol adaptation, and the Application Layer for specific data payload processing. These modules are summarized in Table 4 and Table 5 , respectively.

Table 4. Standard Stream Processing Modules for the FPGA Transport Layer in D-Matrix

Node Name	Layer	Input	Output	Functional Description
Multiplexer (Mux)	Transport	AXI4-Stream	AXI4-Stream	Merges multiple streams into a single output port using a scheduling mechanism (e.g., Round-Robin or Fixed-Priority).

Node Name	Layer	Input	Output	Functional Description
Demultiplexer (Demux)	Transport	AXI4-Stream	AXI4-Stream	Used with a Mux, separates a single input stream into multiple output streams.
Split	Transport	AXI4-Stream	AXI4-Stream	Splits a large input data frame into multiple smaller data frames.
Combine	Transport	AXI4-Stream	AXI4-Stream	Combines multiple small input frames belonging to the same large frame into a single complete data frame.
vFIFO	Transport	AXI4-Stream	AXI4-Stream	A multi-channel, large-capacity virtual FIFO implemented using DDR memory.
Backpressure Notif. (BPN)	Transport	AXI4-Stream	AXI4-Stream	Sends a command to the source to pause transmission when the receiving buffer is nearly full.
Backpressure Ctrl. (BPC)	Transport	AXI4-Stream	AXI4-Stream	Pauses data transmission upon receiving a backpressure notification.
Reliability Tx	Transport	AXI4-Stream	AXI4-Stream	Encapsulates data with sequence IDs and CRC checksums, and buffers data for retransmission upon receiving error feedback.
Reliability Rx	Transport	AXI4-Stream	AXI4-Stream	Verifies CRC and sequence continuity, strips headers, and generates feedback commands to request retransmission in case of errors.

Table 5. Standard Stream Processing Modules for the FPGA Application Layer in D-Matrix

Node Name	Layer	Input	Output	Functional Description
Merge [36]	Application	SDMF	SDMF	Merges multiple data frames from a lower-level domain into a single upper-level domain frame.

Node Name	Layer	Input	Output	Functional Description
Regularizer	Application	SDMF	SDMF	Sorts data frames from multiple same-level input ports while preserving the frame structure.
Trigger	Application	SDMF	SDMF	Receives external triggers, opens a time window, and allows data within this window to pass through.
Feature Extractor	Application	SDMF	SDMF	Performs a two-operand operation on incoming stream data and stored values, updating the storage. Used for online information extraction.
Command Router [33]	Application	SDMF	SDMF	Manages auto-generation of physical node addresses, command routing, and local command filtering.
Cmd. Feedback Collector	Application	SDMF	SDMF	Collects command feedback (e.g., read results) and propagates it up the hierarchy.
Broadcast	Application	SDMF	SDMF	Sends data frames from a single input port to multiple output ports.
Multiplexer (Mux)	Application	SDMF	SDMF	Outputs frames from multiple input ports to a single output port based on a priority scheme.
Map_T	Application	SDMF	SDMF	Distributes data from an input port to multiple output ports using the time index as a hash key.
Bus Latch	Application	SDMF	SDMF	Provides SDMF-based data buffering. Registers performance.
Auto-Expander	Application	SDMF	SDMF	Automatically pads the input SDMF with the high-order time index or spatial addresses based on the port number.

Figure 10

Figure 2: Figure 10

Node Name	Layer	Input	Output	Functional Description
Decoder	Transport / Application	AXI4-Stream	SDMF	Decodes D-Matrix standard frames from the AXI4-Stream interface to the SDMF interface.
Encoder	Application / Transport	SDMF	AXI4-Stream	Encodes SDMF signals into D-Matrix standard frames for transmission via the AXI4-Stream interface.
Blackhole	Application	SDMF	-	Acts as a data sink for simulation or testing, with a configurable reception rate.

C. DAQ Interface IP Core

Regarding the interface between the DAQ and detector subsystems, the CEE DAQ adopts the strategy of embedding a DAQ Interface IP Core directly into the subsystem FEEs. This design approach not only facilitates integration for subsystem users but also ensures strict adherence to the interface protocols, thereby minimizing potential integration errors. Furthermore, unified management of the transmission logic at both ends of the optical link significantly facilitates link commissioning and system maintenance.

As illustrated in Fig. 9 [FIGURE:9], the DAQ Interface IP Core primarily encapsulates transport and command-related modules, exposing three interfaces to the FEE user logic: - **Standard AXI4-Stream Interface**: Utilized for the transmission of Data, Status, and Urgent streams. - **Extended AXI4-Stream Interface**: Equipped with additional address lines, dedicated to the transmission of Command and Command Feedback streams. - **Global Register Access**: Provides access to system attributes (e.g., the allocated FEE ID) and common control signals, such as status upload enables, synchronization checks, and logic resets.

D. CROB-PXI Board

The CROB-PXI board serves as a specific variant of the CROB within the D-Matrix platform, designed based on the PXI mechanical and electrical specifications. It adheres to the standard 6U form factor, with dimensions of 233.35 mm $\times$ 160.00 mm.

1. Board Design and Components As illustrated in Fig. 10

, the CROB-PXI board utilizes a Kintex UltraScale FPGA (XCKU040-FFVA1156-2I) as the main processing core, responsible for managing all on-board modules and data interfaces. It features 16 high-speed serial GTH transceivers connected to the FMC interface for data acquisition from front-end FEEs, and two GTH transceivers linked to SFP+ optical interfaces for data uplink to the upper-level CROB-PCIe. Two high-capacity DDR4 memory chips are integrated for data buffering.

In addition to the main processing unit, the board incorporates several auxiliary systems. LEMO connectors are connected to the Main FPGA to serve as a backup interface for external triggers. An Artix-7 FPGA (XC7A100T-CSG324-2I) acts as the auxiliary controller, managing PXI backplane signals (reserved for future use). The board also utilizes an STM32 microcontroller for system initialization and configuration of I2C-based peripherals, including clock management, temperature monitoring, and EEPROM-based board identification. The power supply system supports a dual-source selection mechanism, allowing power input from either the PXI chassis or an external source.

2. Stream Processing Logic The CROB-PXI board serves as the hardware entry point for the DAQ system. A simplified schematic illustrating the deployment of its stream processing modules is presented in Fig. 11 [FIGURE:11].

Regarding the data stream, the CROB-PXI is tasked with first-level aggregation. Incoming data frames from multiple FEEs are first processed by a Decoder, which converts them from AXI4-Stream into SDMF streams. Subsequently, these streams pass through the Merge module to undergo first-level sub-event merging, forming aggregated board-level data. Finally, the data is converted back to standard AXI4-Stream via an Encoder and transmitted to the upper-level CROB-PCIe board.

Beyond the primary data path, the board handles auxiliary streams. The data stream from FEEs passes through a Flowmeter to gather traffic statistics, which are subsequently packaged into a status frame by the Monitor Buffer for uplink transmission. Urgent messages are also generated locally on the board. Since the CROB-PXI performs no logic processing on status or urgent messages, these locally generated streams are combined with the corresponding streams received from downstream FEEs via a Multiplexer (Mux) and transmitted to the upstream CROB-PCIe. Both the command and command feedback streams are managed by the Command Router module. For the downstream command stream, the router examines the destination address of each command frame to determine whether to execute the command locally or forward it to the corresponding FEE. In contrast, the command feedback stream undergoes no local processing; it is simply collected by the Command Router and forwarded upstream.

While the firmware logic architecture of the CROB-PXI remains uniform across different detector subsystems, specific parameter configurations of the Merge

Figure 12

Figure 3: Figure 12

module vary to accommodate the distinct spatiotemporal characteristics of the scientific data from each detector.

E. CROB-PCIe Board

The CROB-PCIe board represents another variant of the CROB within the D-Matrix platform, implemented based on the PCIe Gen 2.0 x8 interface standard. Mechanically, it adheres to the half-length, full-height add-in card form factor defined in the PCIe Gen 2.0 specification, with dimensions of 167.65 mm × 111.15 mm.

1. Board Design and Components As illustrated in Fig. 12

, the CROB-PCIe board utilizes a Xilinx Kintex-7 FPGA (XC7K325T-FFG900-2I) as the central processing unit, responsible for coordinating the entire data transmission workflow and logic interactions. The FPGA integrates 16 high-speed serial GTX transceivers: 8 are routed to the FMC connector to receive data from the downstream CROB-PXI, while the remaining 8 are connected to the PCIe edge connector for data uplink to the host server. DDR3 memory modules are implemented for data buffering. The board also incorporates an STM32 microcontroller to manage system initialization and configure I2C-based peripherals during the power-up phase. In terms of connectivity, the board features a USB 3.0 interface reserved for future functional expansions and external device interconnections.

2. Stream Processing Logic Serving as the interface between the readout electronics system and the back-end server cluster, the CROB-PCIe board is installed within the DAS. A simplified block diagram illustrating its firmware module deployment is presented in Fig. 13 [FIGURE:13]. The processing mechanisms for command, command feedback, status, and urgent message streams on the CROB-PCIe are fundamentally consistent with those on the CROB-PXI. Regarding the data stream, the CROB-PCIe is currently configured in a transparent transmission mode, where aggregated board-level sub-events packaged by the CROB-PXI are forwarded directly to the DAS.

F. FPGA Mezzanine Card

Within the D-Matrix platform, these CROB boards are designed in adherence to the FPGA Mezzanine Card (FMC) standard [45], where the carrier board provides computational resources and the mezzanine module provides interface resources. Functioning as the carrier board, the CROB establishes direct connections between its FPGA I/O pins and the I/O signals of the mezzanine module

via high-density FMC connectors. This design paradigm offers exceptional flexibility, allowing the CROB to adapt to variations in FEE transmission interfaces or deployment scales by simply swapping the mounted mezzanine module. This capability enables the flexible adjustment of transmission media and the expansion of the number of interfaces.

Fig. 14 [FIGURE:14] presents several FMC mezzanine modules commonly employed within the D-Matrix platform. Specifically, the FM-S18 and FM-S14 are transmission-oriented modules manufactured by Fast Technology, featuring 8 and 4 SFP optical interfaces, respectively. The FMC-QSFP is a custom in-house design incorporating two QSFP interfaces and one RJ45 Ethernet port, characterized by its lower physical profile and higher integration density. Additionally, the QT7426A, produced by Queentest, utilizes two QSFP interfaces for optical data transmission.

V. Software Design

This section presents the software design of the CEE DAQ and describes the processing mechanisms for various information streams.

A. Software Module Interface

The core processing components of the CEE DAQ software are primarily developed in C++. To support this modular architecture, precise interface definitions are essential.

Within a server, driven by the modular design strategy, individual software modules operate as independent processes. Consequently, the module interface must be designed to facilitate Inter-Process Communication (IPC). The SDMS, serving as the standard software interface in the D-Matrix platform, features a core architecture combining a message queue with a dual shared memory scheme (Fig. 15 [FIGURE:15]). Data transmission based on shared memory represents the most efficient IPC mechanism. By mapping the same physical memory region into the user spaces of different processes, data can be transferred between processes with zero-copy overhead, while significantly bypassing the need for kernel system calls.

Further details regarding the design of SDMS can be found in [34].

Data transmission between nodes is classified into two categories. The first category encompasses hardware-to-software data transfer and final data storage. In the Linux environment, these interactions are abstracted as file operations; consequently, they are implemented directly using standard file descriptor read/write functions. The second category pertains to inter-server data interaction, for which the CEE DAQ utilizes ZeroMQ [46] to handle all network-based data exchange.

B. Standard Software Stream Processing Modules

Similar to the firmware modules, D-Matrix provides a suite of standard software module implementations, as summarized in Table 6 . Notably, several key modules are available in both firmware and software versions.

Table 6. Standard Stream Processing Modules for the Software Layer in D-Matrix

Node Name	Input	Output	Functional Description
FD_{Receive}	File Descriptor	Socket	Reads standard D-Matrix data frames from a file descriptor.
FD_{Transmit}	Socket	File Descriptor	Writes standard D-Matrix data frames from an SDMS stream to a file descriptor.
ZMQ_{Receive}	Socket	Socket	Reads standard D-Matrix data frames from a ZeroMQ socket.
ZMQ_{Transmit}	Socket	Socket	Writes standard D-Matrix data frames from an SDMS stream to a ZeroMQ socket.
Command Hub	Socket	Socket	Collects command streams from multiple sources, merges them into a single channel, and routes feedback to the originating source.
Command Router [33]	Socket	Socket	Manages auto-generation of physical node addresses, command routing, and local command filtering.
Cmd. Feedback Collector	Socket	Socket	Collects command feedback (e.g., read results) and propagates it up the hierarchy.
Map	Socket	Socket	Provides three map modes: Broadcast, Time-based (Map_T), and Space-based (Map_S).
Merge [36]	Socket	Socket	Merges multiple data frames from a lower-level domain into a single upper-level domain frame.
Project	Socket	Memory	Projects data into a memory buffer indexed by channel address, removing the time dimension. Output is triggered by a timer or pull request.

Node Name	Input	Output	Functional Description
Snapshot	Socket	External	Provides the latest data frame from the input for external access. The frame is locked during access and updated upon release.
Save	Socket	File Descriptor	Saves incoming D-Matrix data frames to disk, optionally with buffering. Can act as a stream endpoint.
Display	File Descriptor	Socket	Loads D-Matrix data frames from disk and streams them through an SDMS interface.
Console	Socket	GUI	Visualizes a stream of D-Matrix frames in a 2D/3D GUI (Qt or web-based). Can act as a stream endpoint. Sends auto-numbering commands, detects and displays system hardware, and handles urgent messages for the GUI.
Fabric	Socket	Socket	A data distribution module that implements an M-to-N communication network. It is composed of several submodules and supports multiple task scheduling strategies.
Blackhole	Socket	-	Acts as a stream endpoint that consumes data infinitely with zero latency. Used for testing purposes.

C. Scientific Data Processing

Scientific data processing constitutes one of the primary responsibilities of DAQ systems in large-scale physics experiments, encompassing data transmission and aggregation, event building, and the storage of final event files.

1. Event Building Serving as the core engine for event building, the Merge module in the CEE DAQ features a generic design, whose merging behavior is illustrated in Fig. 16 [FIGURE:16]. The key configuration parameters for this module are the spatiotemporal attributes describing the input and output data spaces. The Merge module determines its merging behavior by comparing the disparities in these spatiotemporal attributes. Detailed merge rules and algorithms are described in [36].

The CEE DAQ implements a two-level merging hierarchy. The first level is

executed on the CROB-PXI board, which merges data from 8 or 16 FEEs to generate aggregated board-level sub-events. The second level takes place in the EBS within the server cluster, where all board-level data is collected to complete full event building. Although it is technically feasible to configure additional intermediate sub-event merging stages on the CROB-PCIe and DAS, the current two-level architecture sufficiently satisfies the experimental requirements of the CEE.

2. Fabric Full event building requires aggregating data from all channels associated with a specific Trigger ID. Given that both the DAS and EBS operate as distributed clusters, this process necessitates a precise data routing mechanism. Specifically, all data fragments corresponding to a unique Trigger ID must be gathered from the various DAS nodes and dispatched to a designated target EBS node for final assembly.

To implement a fully connected switching network within the cluster, the CEE DAQ employs the Fabric module. The Fabric (Fig. 17 [FIGURE:17]) functions as an abstract module composed of multiple concrete submodules, specifically including a Fabric Manager, multiple Fabric Transmitters, and multiple Fabric Receivers.

Crucially, the Fabric module supports multiple task scheduling strategies, including weighted rule distribution and minimum-load scheduling. It integrates fault-tolerant scheduling mechanisms to automatically reroute data upon node failure, and supports dynamic node addition to allow seamless expansion of processing resources during operation. Coupled with comprehensive status monitoring, this design ensures that the event building network is highly scalable and can be dynamically adjusted according to the experiment' s scale and requirements.

3. Save The EBS servers are individually connected to a commercial NAS cluster via 10 Gbps Ethernet interfaces, utilizing the Network File System (NFS) mounted in asynchronous mode. For data persistence, the Save module invokes Boost.Filesystem to execute file write operations.

Given that data frames exist as structured entities in memory but require conversion into a continuous serial stream for file storage, this operation essentially functions as an encoding (or serialization) process. To maximize efficiency, circular buffers are employed in both decoder and encoder processes, ultimately decoupling the processing threads from the I/O transmission threads.

D. Run Control and Management

1. Component Management Component management and run control for the entire experiment constitute another critical responsibility of the DAQ system. Given the hierarchical and aggregate nature of the experiment, system components inherently form a tree topology. Consequently, the CEE DAQ

incorporates a generic command system that features automatic node traversal. This mechanism yields routing-based identification results, effectively supporting clustering and hierarchical node management requirements, as detailed in [33]. A visualization of the actual CEE DAQ system topology, automatically generated through this traversal mechanism, is presented in Fig. 18 [FIGURE:18].

2. Run Control The CEE DAQ provides a general-purpose main control software suite. The core of this generic control software is developed primarily in Python and consists of two main components: core functional groups and executors (Fig. 19 [FIGURE:19]).

The core functional groups encapsulate essential system control functionalities, including user management, subsystem configuration (which interacts with the database), and system operational control. Given that the underlying D-Matrix system is predominantly developed in C++, the operational control logic requires seamless interoperability with C++ components. This interaction is facilitated through encapsulated interfaces utilizing ZeroMQ and Boost.Python. Furthermore, the system ensures high extensibility by automatically scanning and registering Python modules located in designated directories.

The architecture defines three types of core executors: a terminal console, a script executor, and a web server. Notably, all three executors invoke functionalities via a unified command format: `<Group> <Command> [Parameters]`.

The terminal console supports both interactive sessions and single-command execution, while the script executor handles predefined script files. The web server, implemented using the Flask framework [47], interfaces with front-end web pages to perform operations and retrieve data using the same command structure. This diversity of executors provides the control software with significant versatility, catering to a wide range of application scenarios.

Furthermore, regarding experimental operations, the D-Matrix orchestrates the system workflow into distinct operational phases, such as initialization, parameter configuration, synchronization, acquisition start/stop, and global reset. The specific control behaviors and configuration logic for individual detectors within each phase are encapsulated in modular Python scripts. Developed by the subsystem designers, these scripts implement device-specific operations primarily through register read/write commands. Crucially, this architecture allows scripts for different operational modes to be configured and modified independently, enabling dynamic runtime switching based on mode selection. Additionally, internal DAQ functionalities also leverage this script-based approach to enhance flexibility for routine tasks.

In addition, the CEE DAQ provides a web-based front-end interface developed using Vue.js [48] and Element Plus [49]. A representative screenshot of this user interface is illustrated in Fig. 20 [FIGURE:20].

E. Status Monitoring

The CEE experiment incorporates a dedicated SCS [41], which is responsible for the display and archiving of detector status monitoring, as well as the supervision and control of auxiliary systems such as High Voltage (HV) supplies and gas systems. However, since all status data regarding the readout electronics is transmitted upstream via the DAQ links, a dedicated status interface must be established to forward this information from the DAQ to the SCS.

The interface between the DAQ and the SCS is implemented using the Experimental Physics and Industrial Control System (EPICS) [50], a set of open-source software tools and applications widely adopted for creating distributed soft real-time control systems in large-scale scientific facilities.

Specifically, the DAQ system instantiates an EPICS Input/Output Controller (IOC) to bridge the data exchange. Within this IOC, a global status Process Variable (PV) is defined for each entity type across the subsystems. In this context, entities sharing identical firmware designs and status definitions are categorized as the same type. Upon receiving status frames from the readout electronics, the DAS forwards them directly to the OCS. The OCS identifies the target PV based on the board type information within the frame. It then calculates the starting offset address for the specific board within the aggregated PV—derived from the board ID and the predefined status data size for that board type—and updates the PV accordingly.

F. Error Handling and Logging

The CEE DAQ implements two distinct logging architectures. The C++ components utilize the log4cplus library [51], where log management is organized primarily on a per-server basis. INFO-level logs from individual processes are stored in local rotating files. In contrast, logs at the WARNING-level or higher are aggregated into a unified server-level rotating file and simultaneously transmitted to the OCS via a custom-designed ZeroMQ Appender.

On the OCS side, the main control software employs Python' s built-in logging module, similarly recording local INFO-level data into rotating files. Furthermore, the control software receives the forwarded C++ logs via the ZeroMQ interface and pushes the aggregated log streams to the front-end web interface for real-time visualization.

Urgent message streams originating from the readout electronics are received by individual DAS nodes and forwarded to the OCS. Upon receiving these messages, the OCS first identifies the associated detector subsystem based on the board type information. It then invokes the specific urgent message handler function—defined within the subsystem' s registered script—to execute the necessary error handling procedures.

VI. System Integration and Performance

This section describes the system setup and presents key performance test results.

A. System Setup

By June 2025, the assembly of the CEE spectrometer was completed, with the exception of the dipole magnet (Fig. 21 [FIGURE:21]). Subsequently, two field-off experiments were conducted in July and December of the same year, respectively. The completed on-site hardware deployment of the CEE DAQ system, spanning from the PXI-based readout aggregation layer to the back-end server cluster, is illustrated in Fig. 22 [FIGURE:22].

The front-end aggregation layer consists of 32 CROB-PXI boards. To accommodate the varying data bandwidths of different detectors, these boards support flexible channel grouping strategies. For low-rate subsystems (e.g., TOF and ZDC), the 16 downlink input channels are aggregated into a single 10 Gbps uplink. In contrast, for the TPC subsystem, which accounts for the largest portion of the system' s total data bandwidth, the board is configured into split mode (two groups of 8 inputs) utilizing dual uplinks to prevent bottlenecks. Consequently, the readout links from the TPC are distributed across multiple back-end servers to ensure effective load balancing.

The back-end computing infrastructure comprises a cluster of 13 high-performance servers. To interface with the optical uplinks, 10 CROB-PCIe cards are deployed, installed within 10 designated nodes of the 12 Nettrix R620 G40 computing servers. This hardware configuration defines the logical roles within the cluster: the 10 nodes hosting PCIe cards function as Data Access Servers (DAS), while the entire 12-node cluster functions collectively as the Event Building Server (EBS) pool. In actual deployment, DAS and EBS processes run concurrently on these nodes to maximize resource utilization.

Each computing node is equipped with dual Intel Xeon Gold 5317 processors (12 cores, 3.0 GHz) and 64 GB of DDR4-3200 memory. To meet high-throughput requirements, these servers feature standard dual-port 10 Gb Ethernet Network Interface Cards (NICs) configured to isolate traffic: one link serves the Fabric for internal data exchange, while the other is dedicated to the storage network for high-speed data recording to the NAS. For system orchestration, a Dell PowerEdge R750 serves as the dedicated Online Control Server (OCS). It is configured with dual Intel Xeon Gold 6330 processors (28 cores, 2.0 GHz) to ensure robust processing power for global system management.

B. Transmission Performance

To validate the data transmission capability of the CEE DAQ system, throughput tests were conducted across all critical physical interfaces along the data path. The evaluation scope encompasses the optical links between the FEE

and the CROB-PXI, the uplink optical connections from the CROB-PXI to the CROB-PCIe, the PCIe 2.0 x8 interface connecting to the host server, and the 10 Gb Ethernet for inter-server communication.

The test results, comparing the theoretical bandwidths with the measured sustained throughputs, are summarized in Table 7. As indicated, all interfaces demonstrated a bandwidth utilization exceeding 90%, confirming the high efficiency and reliability of the transmission architecture design.

Table 7 . Transmission performance test results of key DAQ interfaces

Connection	Interface Type	Theoretical (Gbps)	Measured (Gbps)	Utilization
FEE to CROB-PXI	Optical Fiber	-	-	96.8%
CROB-PXI to CROB-PCIe	Optical Fiber	-	-	96.8%
CROB-PCIe to Server	PCIe Gen 2.0 x8	-	-	90.6%
Inter-Server	10 Gb Ethernet	-	-	94.0%

C. Module-Level Performance

1. Firmware Event Building Performance First-level event building was evaluated using an internal data generator on the CROB-PXI board, configured to mimic the high-bandwidth TPC subsystem (merging 8 channels with 4117 Byte frames). The firmware merge module achieved an event building rate of 22 028 Hz, corresponding to a throughput of 725.55 MB/s. This performance significantly exceeds the theoretical peak load for a single TPC split-group, validating the efficiency of the FPGA-based aggregation logic.

2. Software Event Building Performance Second-level event building was benchmarked on a single server node functioning simultaneously as a data generator (DAS) and an event builder (EBS, without disk storage) to isolate processing performance. Targeting the CEE full-scale requirement (4.5 GB/s at 10 kHz, approx. 450 KB/event), the single-node performance with 4 worker threads reached 3937 Hz and 1.69 GB/s. Given the deployment of 12 EBS nodes, the cluster possesses sufficient aggregate processing capacity to accommodate the full system load.

3. Data Storage Performance Storage performance was evaluated on a single node writing to the NAS with aggregation optimized to 10 000 frames/file. As illustrated in Fig. 23 [FIGURE:23], throughput increases with frame size due to reduced I/O overhead. At a frame size of 400 KB (close to the average CEE event size), the single-node write speed reaches 845.7 MB/s using 8 threads. This confirms that the distributed storage system can robustly handle the required data bandwidth.

D. System-Level Performance

To verify the overall capability of the CEE DAQ cluster under high-load conditions, integrated system-level performance tests were conducted, focusing on event building and data storage.

1. Full-System Event Building The aggregate event building capacity was evaluated using software-simulated data sources configured to mimic the realistic CEE data distribution. In this setup, consistent with the physical hardware layout, 10 nodes functioned as DAS (generating data) while the entire 12 server cluster operated as EBS (performing assembly with 8 worker threads per node).

Each DAS node simulated 4 data sources, comprising two large streams (simulating TPC data at 23 040 Bytes) and two small streams (simulating other subsystems at 135 Bytes). Under this heterogeneous payload configuration, the test yielded an aggregate event building rate of 27 202 Hz, corresponding to a total system throughput of 11.74 GB/s. This performance significantly surpasses the design requirement of 4.5 GB/s, confirming that the back-end cluster possesses sufficient processing headroom even under complex realistic data patterns.

2. Cluster Storage Performance The storage performance of the entire cluster was assessed with all 12 nodes executing concurrent write operations to the NAS. The aggregated throughput results under different configurations are illustrated in Fig. 24 [FIGURE:24]. As shown, the cluster storage performance scales effectively with frame size. At a frame size of 400 KB with 8 threads per node, the peak aggregate throughput reaches 6880.9 MB/s (approx. 6.72 GB/s). This performance well exceeds the steady-state storage requirement of 900 MB/s, ensuring reliable data recording during full-load experiments.

VII. Conclusion

As the first large-scale nuclear physics experimental facility in China operating in the GeV energy region, the CEE imposes stringent requirements on data acquisition performance and system reliability. In this paper, we have presented the complete design and implementation of the CEE DAQ system. The system

is constructed based on D-Matrix, a firmware-software co-designed stream processing platform. This represents the first large-scale engineering application of the D-Matrix architecture. By adopting a modular stream processing paradigm, the system realizes a unified design methodology for both firmware and software, resulting in a highly flexible and scalable architecture capable of adapting to diverse detector subsystems.

Comprehensive performance evaluations validate the robustness of the system. In full-system tests, the DAQ cluster achieved an aggregate event building rate of 27 202 Hz with a throughput of 11.74 GB/s, and a peak storage throughput of 6.72 GB/s. These performance metrics far exceed the experimental requirements of the CEE. Furthermore, the system' s reliability and stability have been practically validated through its successful operation during three module-level beam integration tests and two field-off spectrometer experiments.

The successful implementation and on-site deployment of the CEE DAQ system demonstrate the efficacy of the proposed architecture. The technologies and design methodologies verified in this work will serve as a foundational reference for future large-scale physics facilities, including the High Energy Fragment Separator (HFRS) [52] and the Super Tau-Charm Facility (STCF) [53].

References

[1] L. Lü, H. Yi, Z. Xiao, et al., Conceptual design of the HIRFL-CSR external-target experiment. Science China Physics, Mechanics & Astronomy, 60, 012021 (2017). doi: 10.1007/s11433-016-0342-x

[2] J. W. Xia, W. L. Zhan, B. W. Wei, et al., The heavy ion cooler-storage-ring project (HIRFL-CSR) at Lanzhou. Nuclear Instruments and Methods in Physics Research Section A, 488, 11–25 (2002). doi: 10.1016/S0168-9002(02)00475-8

[3] Z. Xiao, L. W. Chen, F. Fu, et al., Nuclear matter at a HIRFL-CSR energy regime. Journal of Physics G: Nuclear and Particle Physics, 36, 064040 (2009). doi: 10.1088/0954-3899/36/6/064040

[4] Y. J. Yuan, J. C. Yang, J. W. Xia, et al., Status of the HIRFL–CSR complex. Nuclear Instruments and Methods in Physics Research Section B, 317, 214–217 (2013). doi: 10.1016/j.nimb.2013.07.040

[5] X. Luo, S. Shi, N. Xu, et al., A study of the properties of the QCD phase diagram in high-energy nuclear collisions. Particles, 3, 278–307 (2020). doi: 10.3390/particles3020022

[6] Z. Hu, C. Gao, X. Tian, et al., A low noise 2.2 GHz PLL ASIC for the CSR external-target experiment. Journal of Instrumentation, 18, C02011 (2023). doi: 10.1088/1748-0221/18/02/C02011

[7] J. Liu, C.-S. Gao, H.-L. Wang, et al., Readout electronics for beam monitor in the external-target experiment of CSR. Nuclear Science and Techniques, 36, 87 (2025). doi: 10.1007/s41365-025-01652-0

[8] H. Wang, Z. Wang, C. Gao, et al., A gaseous beam monitor for the CSR external-target experiment. Physics of Atomic Nuclei, 86, 1411–1415 (2023). doi: 10.1134/S1063778823060212

[9] H.-L. Wang, Z. Wang, C.-S. Gao, et al., Design and tests of the prototype beam monitor of the CSR external target experiment. Nuclear Science and Techniques, 33, 36 (2022). doi: 10.1007/s41365-022-01021-1

[10] B. You, C. Gao, P. Yang, et al., The Topmetal-CEE prototype, a direct charge sensor for the beam monitor of the CSR external-target experiment. Journal of Instrumentation, 17, C09030 (2022). doi: 10.1088/1748-0221/17/09/C09030

[11] J. Cao, J. Lu, L. Zhao, et al., Prototype design of the time measurement circuit for SiPM-based T0 detector of CEE. 2022 7th International Conference on Electronic Technology and Information Science (ICETIS), 1–6 (2022). doi: 10.1109/ICETIS55134.2022.9788592

[12] P. Deng, L. Zhao, J. Lu, et al., Readout electronics of T0 detector in the external target experiment of CSR in HIRFL. IEEE Transactions on Nuclear Science, 65, 1315–1323 (2018). doi: 10.1109/TNS.2018.2834426

[13] D. Hu, X. Wang, M. Shao, et al., Design and performance testing of a T0 detector for the CSR external-target experiment. Nuclear Instruments and Methods in Physics Research Section A, 1057, 168773 (2023). doi: 10.1016/j.nima.2023.168773

[14] L. He, Z. Song, F. Lu, et al., Simulation of momentum resolution of the CEE-TPC in HIRFL. Nuclear Techniques, 39, 70401 (2016). doi: 10.11889/j.0253-3219.2016.hjs.39.070401

[15] W. Huang, F. Lu, H. Li, et al., Laser test of the prototype of CEE time projection chamber. Nuclear Science and Techniques, 29, 41 (2018). doi: 10.1007/s41365-018-0382-4

[16] J. Yuan, Y. Qian, H. Zhao, et al., Development of multichannel readout electronics prototype system for TPC detector of CSR external-target experiment. Nuclear Instruments and Methods in Physics Research Section A, 1052, 168281 (2023). doi: 10.1016/j.nima.2023.168281

[17] J. Lu, L. Zhao, J. Qin, et al., Readout electronics prototype of TOF detectors in CEE of HIRFL. IEEE Transactions on Nuclear Science, 68, 1976–1983 (2021). doi: 10.1109/TNS.2021.3093544

[18] X. Wang, D. Hu, M. Shao, et al., CEE inner TOF prototype design and preliminary test results. Journal of Instrumentation, 17, P09023 (2022). doi: 10.1088/1748-0221/17/09/P09023

[19] Y. Zhou, D. Hu, M. Shao, et al., Mass production and quality control of CEE iTOF-MRPC. Nuclear Instruments and Methods in Physics Research Section A, 1083, 171186 (2026). doi: 10.1016/j.nima.2025.171186

[20] Y. Zhou, D. Hu, X. Wang, et al., R & D of prototype iTOF-MRPC at CEE. Nuclear Instruments and Methods in Physics Research Section A, 1054, 168455 (2023). doi: 10.1016/j.nima.2023.168455

[21] Z.-B. He, Z. Qin, P. Ma, et al., Development of a MWDC prototype of the CSR external-target experiment. Nuclear Science and Techniques, 35, 174 (2024). doi: 10.1007/s41365-024-01439-9

[22] L.-M. Lyu, H. Yi, L.-M. Duan, et al