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: 2026-04-07T13:16:01+00:00

Abstract

The Cooling Storage Ring (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 Common ReadOut Boards (CROBs), while the software stack supports dynamic global control and high-performance data handling. System-level performance evaluations demonstrate sustained throughputs of 12.61 GB/s for aggregate event building (at 27202 Hz) and 7.21 GB/s for cluster storage. 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 High Energy Fragment Separator (HFRS) and Super Tau-Charm Facility (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

Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China

The Cooling Storage Ring (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 Common ReadOut Boards (CROBs), while the software stack supports dynamic global control and high-performance data handling. System-level performance evaluations demonstrate sustained throughputs of 12.61 GB/s for aggregate event building (at 27,202 Hz) and 7.21 GB/s for cluster storage. 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 the High Energy Fragment Separator (HFRS) and Super Tau-Charm Facility (STCF).

Keywords

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

INTRODUCTION

The CEE [?] 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) [?, ?, ?], 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) [?]. shows the main technical performance of the CEE.

As shown in [Figure 1: see original paper], the CEE spectrometer comprises a dipole magnet and a suite of detector systems, including a beam monitor [?, ?, ?, ?, ?], a start time detector (T0) [?, ?, ?], a time projection chamber (TPC) [?, ?, ?], an inner time-of-flight detector (iTOF) [?, ?, ?, ?], a multi-wire drift chamber (MWDC) [?, ?, ?, ?, ?], an external time-of-flight detector (eTOF) [?, ?, ?, ?, ?], and a zero-degree calorimeter (ZDC) [?, ?, ?]. Additionally, the CEE incorporates essential common subsystems, including the DAQ [?, ?, ?, ?], trigger [?], clock [?, ?, ?], and Slow Control System (SCS) [?].

In modern large-scale physics experiments, DAQ systems are increasingly transitioning into distributed, heterogeneous computing architectures. To manage the growing complexity, contemporary DAQ solutions often adopt standardization strategies at specific layers. For instance, implementations like GigaBit Transceivers (GBT) [**?**] and Front-End LInk eXchange (FELIX) [**?**] have successfully standardized hardware data links for transparent data transportation. In the software domain, frameworks such as artdaq [**?**], xDAQ [**?**], and FairMQ [**?**] define common data containers and provide a unified approach to constructing generic data processing units. However, a common limitation in these architectures is that Field Programmable Gate Array (FPGA) firmware and Central Processing Unit (CPU) software are typically treated as disjointed development domains. This hardware-software separation restricts the flexible distribution of processing tasks and complicates the iterative development cycle.

The CEE spectrometer comprises highly heterogeneous detector subsystems with different data rates and payload structures. More importantly, as a large-scale engineering facility, its construction spans a multi-phase lifecycle from initial laboratory-scale Front-End Electronics (FEE) evaluations and sub-system joint testing to the final full-array on-site integration. These practical factors impose demands on the DAQ system's scalability and lifecycle adaptability, minimizing the need for extensive logic redesigns when migrating across different deployment phases.

To address these distinct requirements, the CEE DAQ system is implemented based on D-Matrix [**?**, **?**, **?**, **?**, **?**], a generic firmware-software co-designed stream processing platform. By establishing a unified design methodology across both hardware and software domains, D-Matrix breaks the traditional development barriers. Its modular stream processing paradigm allows identical logical designs to be highly reusable and seamlessly scaled across different hardware deployment phases. This paper presents the comprehensive design of the CEE DAQ system, involving in-depth research into distributed architecture design, protocol specification, stream processing modules, and performance optimization, demonstrating how this unified approach effectively fulfills the complex engineering demands of the CEE.

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 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 . Specifically, the real-time event building capacity is derived from the maximum design event rate of the spectrometer presented in . The data throughput specification is derived from the estimated peak data rates of the individual detector systems, as presented in .

In terms of storage specifications, the CEE experiment utilizes a beam with slow extraction, resulting in a periodic time structure (as shown in [Figure 2: see original paper]). 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.

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 [Figure 3: see original paper]. 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 CROBs: the CROB-PXI (Common ReadOut Board on PXI) and the CROB-PCIe (Com-

mon ReadOut Board on PCIe).

The CROB-PXI is installed in a PCI eXtensions for Instrumentation (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 Peripheral Component Interconnect Express (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 entry point for data into the server cluster. It receives data from the CROB-PCIe and distributes it to the EBS cluster 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 [**?**]. 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 independent 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 [Figure 4: see original paper], where individual streams are differentiated by the Stream ID field 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 [**?**].

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 Advanced eXtensible Interface 4 (AXI4)-Stream interface or the custom Standard D-Matrix FPGA (SDMF) interface, whereas software modules utilize the custom Standard D-Matrix Software (SDMS) interface.

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 [Figure 5: see original paper], 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 [Figure 6: see original paper], 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.

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 Advanced Microcontroller Bus Architecture (AMBA) AXI4 protocol [**?**], AXI4-Stream [**?**] 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 Intellectual Property (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 ([Figure 7: see original paper]). 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 [Figure 8: see original paper], 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.

B. Standard Firmware Stream Processing Modules

Based on the modular architecture, a library of standard firmware modules has been implemented and categorized into 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 and , respectively.

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 [Figure 9: see original paper], 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.

1. Board Design and Components As illustrated in [Figure 10: see original paper], the CROB-PXI board utilizes a Kintex UltraScale FPGA (XCKU040-FFVA1156-2I) as the main processing core. It features 16 high-speed serial GTH transceivers connected to the FPGA Mezzanine Card (FMC)

interface for data acquisition from front-end FEEs, and two GTH transceivers linked to Small Form-factor Pluggable Plus (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. The board also utilizes an STM32 microcontroller for system initialization and configuration of I2C-based peripherals.

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 [Figure 11: see original paper]. 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.

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.

1. Board Design and Components As illustrated in [Figure 12: see original paper], the CROB-PCIe board utilizes a Xilinx Kintex-7 FPGA (XC7K325T-FFG900-2I) as the central processing unit. 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. Two high-capacity DDR3 SO-DIMM 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.

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 [Figure 13: see original paper]. 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. FMC Mezzanine Modules

Within the D-Matrix platform, these CROB boards are designed in adherence to the FMC standard [**?**], where the carrier board provides computational resources and the mezzanine module provides interface resources. 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. [Figure 14: see original paper] illustrates several FMC mezzanine modules utilized in the CEE DAQ.

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++. 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 ([Figure 15: see original paper]). Data transmission based on shared memory represents the most efficient IPC mechanism, allowing data to be transferred between processes with zero-copy overhead.

Data transmission between nodes is classified into two categories. The first category encompasses hardware-to-software data transfer and final data storage, implemented using standard file descriptor read/write functions. The second category pertains to inter-server data interaction, for which the CEE DAQ utilizes ZeroMQ [**?**] 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 . Notably, several key modules are available in both firmware and software versions.

C. Scientific Data Processing

Scientific data processing constitutes one of the primary responsibilities of DAQ systems, 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 [Figure 16: see original paper]. 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.

Full event building requires aggregating data from all channels associated with a specific Trigger ID. To implement a fully connected switching network within the cluster, the CEE DAQ employs the Fabric module. The Fabric ([Figure 17: see original paper]) functions as an abstract module composed of multiple concrete submodules, specifically including a Fabric Manager, multiple Fabric Transmitters, and multiple Fabric Receivers.

2. Data Storage 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. 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. A visualization of the actual CEE DAQ system topology, automatically generated through this traversal mechanism, is presented in [Figure 18: see original paper].

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 ([Figure 19: see original paper]). The core functional groups encapsulate essential system control functionalities, including user management, subsystem configuration, and system operational control. The architecture defines three types of core executors: a terminal console, a script executor, and a web server.

Furthermore, 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. A representative screenshot of the web-based user interface is illustrated in [Figure 20: see original paper].

E. Status Monitoring

The CEE experiment incorporates a dedicated SCS [**?**]. 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) [**?**]. Specifically, the DAQ system instantiates an EPICS Input/Output Controller (IOC) to bridge the data exchange.

F. Fault Tolerance, Error Handling, and Logging

1. Fault Tolerance and Error Handling To ensure high availability, the CEE DAQ system implements multi-level fault tolerance. A core design principle is fault isolation. At the data processing level, the Merge module continuously monitors the integrity of incoming data links. It handles Trigger ID anomalies and implements a timeout mechanism to prevent a single stalled link from blocking the entire event building process. At the distributed node level, the Fabric module manages the data exchange network and handles potential server failures through a centralized manager and heartbeat mechanisms. At the hardware level, the system employs an urgent message mechanism to handle critical anomalies like over-voltage or over-temperature alerts.

2. Logging The CEE DAQ implements two distinct logging architectures. The C++ components utilize the log4cplus library [**?**], where logs at the WARNING-level or higher are aggregated into a unified server-level file and transmitted to the OCS. On the OCS side, the main control software employs Python' s built-in logging module and pushes the aggregated log streams to the front-end web interface for real-time visualization.

SYSTEM INTEGRATION, PERFORMANCE, AND RELIABILITY

A. System Setup

By June 2025, the assembly of the CEE spectrometer was completed, with the exception of the dipole magnet ([Figure 21: see original paper]). Subsequently, two field-off experiments were conducted. The completed on-site hardware deployment of the CEE DAQ system is illustrated in [Figure 22: see original paper]. The front-end aggregation layer consists of 32 CROB-PXI boards. The back-end computing infrastructure comprises a cluster of 13 high-performance servers, including 10 DAS nodes and a 12-node EBS pool.

B. Transmission Performance

To validate the data transmission capability, throughput tests were conducted across all critical physical interfaces. The test results, comparing the theoretical bandwidths with the measured sustained throughputs, are summarized in . All interfaces demonstrated a bandwidth utilization exceeding 90%.

C. Module-Level Performance

1. **Firmware Event Building Performance:** The firmware merge module achieved an event building rate of 22,028 Hz, corresponding to a throughput of 725.55 MB/s, which significantly exceeds the theoretical peak load for a single TPC split-group.
2. **Software Event Building Performance:** Second-level event building on a single node reached 3,937 Hz and 1.77 GB/s. Given the deployment of 12 EBS nodes, the cluster possesses sufficient aggregate processing capacity.
3. **Data Storage Performance:** As illustrated in [Figure 23: see original paper], at a frame size of 400 KB, the single-node write speed reaches 886.78 MB/s, confirming the distributed storage system can robustly handle the required bandwidth.

D. System-Level Performance

1. **Full-System Event Building:** Under a heterogeneous payload configuration, the test yielded an aggregate event building rate of 27,202 Hz, corresponding to a total system throughput of 12.61 GB/s. This significantly surpasses the design requirement of 4.5 GB/s.
2. **Cluster Storage Performance:** The aggregated throughput results are illustrated in [Figure 24: see original paper]. At a frame size of 400 KB, the sustained aggregate throughput reaches 7.21 GB/s, well exceeding the steady-state storage requirement of 900 MB/s.

E. System Stability and Fault Tolerance Validation

The system's resilience was tested through dynamic fault injection, including link interruption and node failure. In both scenarios, the system successfully isolated the faults and maintained global operation. A 168-hour continuous stress test was also performed, where the system maintained a stable event building rate without crashes or memory leaks, demonstrating operational maturity.

CONCLUSION

The CEE imposes stringent requirements on data acquisition performance and system reliability. This paper has presented the complete design and implementation of the CEE DAQ system based on the D-Matrix platform. By adopting a modular stream processing paradigm, the system realizes a unified design methodology for both firmware and software. Comprehensive performance evaluations validate the robustness of the system, with an aggregate event building rate of 27,202 Hz (12.61 GB/s) and a storage throughput of 7.21 GB/s. The system' s reliability has been practically validated through successful operation during multiple beam integration tests and field-off experiments. This work will serve as a foundational reference for future large-scale physics facilities, including HFRS [**?**] and STCF [**?**].

AUTHOR DECLARATIONS

A. Declaration of Generative AI Use

During the preparation of this work, the authors used Gemini-3-Pro for the purpose of language editing and grammar checking only. The tool was used to improve the readability and clarity of the text. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article. No AI tool was used to generate scientific data, perform analysis, or formulate conclusions.

Note: Figure translations are in progress. See original paper for figures.