EPJ Web of Conferences **337**, 01201 (2025) https://doi.org/10.1051/epjconf/202533701201
CHEP 2024

High-throughput data distribution for online computing in the CBM experiment

Jan de Cuveland[1,2,*], *Dirk* Hutter[1,2], and *Volker* Lindenstruth[1,2,3]

[1]Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
[2]Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt am Main, Germany
[3]GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

Abstract. The CBM experiment, a fixed-target heavy-ion experiment at the upcoming GSI/FAIR SIS100 facility, aims to investigate QCD at high baryon densities. The CBM First-level Event Selector (FLES) serves as the central event selection system of the experiment. It functions as a high-performance computer cluster tasked with the online analysis of physics data, including full event reconstruction, at an incoming data rate of up to 1 TB/s. The CBM detector systems operate in a free-running and self-triggered manner, delivering time-stamped data streams. Without inherent event separation, timeslice building replaces global event building. The FLES online data distribution integrates data from around 5000 input links into self-contained, overlapping processing intervals and distributes these to the compute nodes. Using a combination of RDMA and zero-copy techniques, timeslices can be built efficiently over a high-throughput InfiniBand network and distributed to available online computing resources for full online event reconstruction and analysis in a heterogeneous HPC cluster system. A new IPC online interface to timeslice data utilizes Posix shared memory governed by a reference-counting item distributor. This design combines maximum performance and flexibility with minimum memory consumption. These developments have been successfully field-tested in production at the CBM predecessor experiment mCBM at the GSI/FAIR SIS18.

1 Introduction

Modern high-energy physics experiments face ever-increasing challenges in data acquisition and processing. The Compressed Baryonic Matter (CBM) experiment, currently being constructed at the Facility for Antiproton and Ion Research (FAIR) in Darmstadt, Germany, represents a particularly demanding case. As a fixed-target heavy-ion experiment, CBM aims to explore the quantum chromodynamics (QCD) phase diagram at high baryon densities through collisions at interaction rates up to 10 MHz [1]. With up to 700 charged particles produced per collision in the detector acceptance, the resulting raw data volume significantly exceeds available storage capacity, necessitating substantial data reduction during experiment operation.

Unlike collider experiments, which typically benefit from well-defined bunch crossing times, CBM employs a quasi-continuous beam on target without inherent event timing structure. The challenge is further amplified by the complex topological signatures of the physics

*e-mail: cuveland@compeng.uni-frankfurt.de

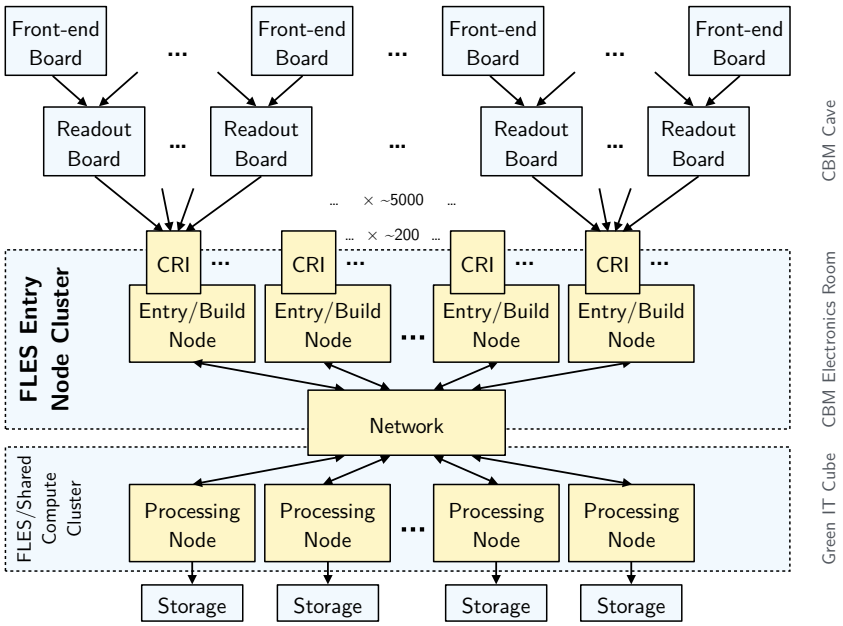

Figure 1. Overview of the online data path architecture of the CBM experiment. The design consists of two clusters, an entry node cluster with the CBM-specific input interfaces, and a shared processing node cluster. To perform timeslice building, the entry nodes and both clusters are connected by a fast network.

processes under study, which require full event reconstruction and analysis for effective event selection. To handle these challenges at high interaction rates, the experiment's architecture implements self-triggered detector readout systems that deliver continuous streams of time-stamped data [2].

This approach eliminates traditional hardware triggers but requires processing the full detector data rate to identify physical interactions and suppress background noise within the continuous data streams. Thus, CBM needs sophisticated near-real-time software processing with guaranteed high throughput capacity to maintain continuous operation. An HPC cluster provides the computing resources required to perform the event reconstruction and selection during experiment operation. The CBM online system is designed to handle peak input data rates of up to 1 TB/s and sustained average rates of 250 GB/s.

This paper presents the design and implementation of CBM's high-throughput data distribution system, which forms the critical interface between detector readout and online computing. We introduce novel approaches to data organization and transfer that enable efficient processing of continuous data streams at substantial rates. The CBM experiment is scheduled for commissioning at FAIR's SIS100 accelerator, with initial operation expected to begin in 2028.

2 System architecture

The data distribution system spans two physically separated computing clusters: an entry cluster located in the CBM service building and a processing cluster housed in FAIR's Green IT Cube data center (cf. Fig. 1). The entry cluster, equipped with custom PCIe input cards,

interfaces directly with detector readout electronics. Designed for input data rates of up to 1 TB/s, it integrates data from around 5000 detector links into self-contained, overlapping *timeslice* processing intervals which are distributed to the online computing resources. The processing cluster, part of FAIR's shared computing resources, receives timeslices and executes online event reconstruction and selection.

An additional challenge arises from the approximately 1 km cable distance between these clusters, requiring careful consideration of network architecture and protocols. The system employs RDMA-enabled InfiniBand networking throughout, with specialized configuration for the long-distance links between clusters.

Efficient data distribution relies on the timeslice building software *Flesnet*, carefully optimized for this high-throughput data collection and dissemination. The Flesnet software is written in modern C++ and is available as open source on GitHub [3].

2.1 Data organization model

Traditional event building approaches, which combine detector data based on trigger decisions or bunch crossing times, are not applicable to CBM's continuous readout scheme. Instead, we introduce a microslice-based data organization model that enables efficient handling of time-stamped data streams while maintaining system flexibility and scalability.

The fundamental challenge stems from the self-triggered nature of CBM's detector systems, which produce streams of hit messages in detector-specific formats. These messages are typically small in size but high in frequency, with the Silicon Tracking System alone generating 32-bit hit messages at rates up to $10\,\mathrm{MHz/cm^2}$. Moreover, these data streams are generally stateful, employing techniques such as epoch messages for timestamp compression. Our model addresses these challenges through a layered approach to data organization.

2.1.1 Microslice concept

Detector data streams are partitioned into microslices—self-contained data containers covering fixed time intervals, typically around 100 µs of experiment time (corresponding to up to 1000 interactions). Each microslice consists of a 32-byte descriptor and a block of experiment data. The descriptor, structured for efficient software access, contains crucial metadata including a global timestamp, equipment identifier, and status flags. This standardized container format enables system-wide data handling without requiring knowledge of detector-specific data formats. The design of the microslice ensures that no contextual information from preceding or subsequent intervals is required to interpret the data, which greatly simplifies subsequent handling.

The microslice concept enables efficient scaling across various running scenarios through dynamic adaptation of the microslice length. For lower data rate configurations, the microslice time interval is increased proportionally, maintaining approximately constant data size per microslice. This approach keeps the fractional overhead constant while reducing network transaction frequency, allowing the system to operate efficiently across multiple orders of magnitude in data rate.

2.1.2 Timeslice building

Microslices are aggregated into timeslice components (TSCs, see Fig. 2), which cover larger time intervals, typically more than 10 ms. Matching timeslice components from multiple input channels are in turn merged to form complete timeslices containing data from the entire detector system for a specific time period (see Fig. 3). These timeslices represent the primary

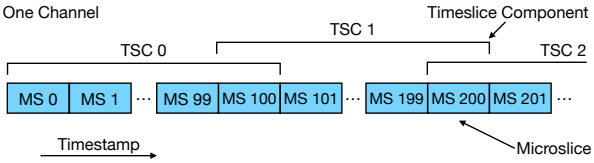

Figure 2. During timeslice building, subsequent microslices are combined to timeslice components, which include a configured overlap.

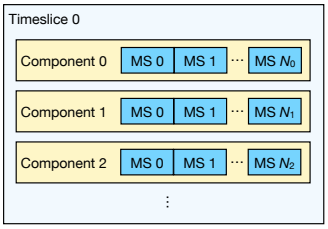

Figure 3. Timeslice components from all sources are finally combined to a timeslice.

data unit for online event reconstruction. The process of timeslice building replaces traditional event building as an organizational step and enables efficient data distribution to the processing cluster.

To address the issue of events that may span the boundaries between adjacent intervals, a controlled overlap is introduced during timeslice building. This overlap is achieved by duplicating a configurable number of microslices at the boundary of successive TSCs. As a result, each timeslice becomes a self-contained data block that is guaranteed to contain complete event information for its core time window, even if some measurements occur near the edges. After the event reconstruction is complete, the interaction time of each event is known with high precision, allowing for accurate association of each event to the core time window of the correct timeslice. As the core time windows do not overlap, the timeslices can be processed independently by the online reconstruction algorithms without the risk of duplicate recognition. This strategy provides an efficient means to ensure that no physics events are lost due to time partitioning while maintaining the ability to process timeslices independently.

The overhead associated with both the container metadata and the intentional duplication for overlap is kept minimal. Detailed studies of the microslice concept indicate that the total fractional overhead remains well below one percent for a wide range of operating parameters. This low overhead is achieved by carefully tuning the microslice size, the overlap length, and the data packing protocols so that network transaction frequency is reduced without sacrificing the needed granularity.

2.2 Zero-copy data flow

Performance optimization focuses on minimizing memory operations through consistent use of zero-copy techniques. The data path employs several complementary mechanisms working in concert. Direct Memory Access (DMA) enables custom input cards to write directly to host memory without CPU intervention. Remote Direct Memory Access (RDMA) provides efficient transfers between compute nodes across the network fabric. These transfers are complemented by shared memory interfaces between software components, completing the zero-copy chain from input through processing.

Input cards write detector data directly to ring buffers in host memory via DMA. These buffers, implemented as POSIX shared memory segments, are accessible to both hardware and software components without additional copying. The same memory regions are registered for RDMA, allowing direct network transfers to receiving nodes. The software framework maintains careful control over buffer management and synchronization while keeping

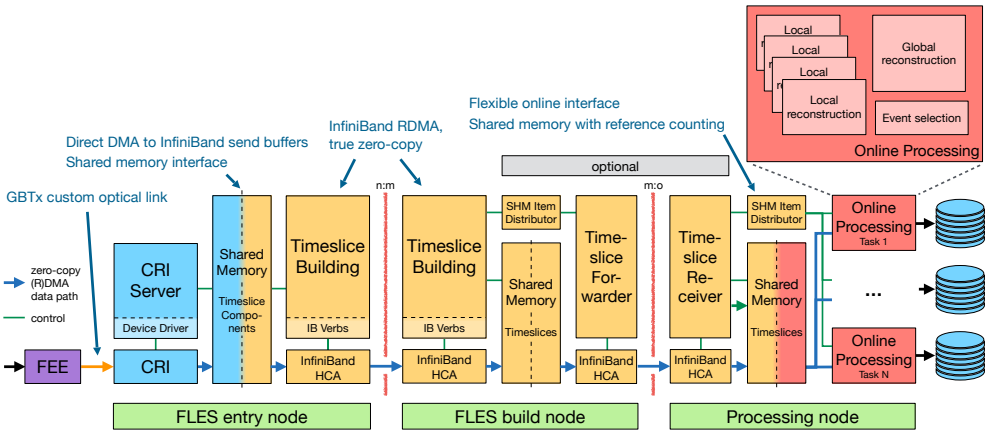

Figure 4. The main components of the CBM online data path. The architecture enables zero-copy data transfer from front-end electronics through entry nodes to processing nodes via a combination of direct memory access (DMA) and remote direct memory access (RDMA) techniques. Shared memory interfaces between software components enable efficient data handling without redundant copying operations.

CPU involvement minimal. Descriptor information flows through separate channels from the main data path, enabling efficient handling of varying data rates and sizes.

2.3 Online interface to timeslice data

To support diverse processing requirements, we developed a flexible interface for accessing timeslice data in the online environment. The interface combines shared memory access for maximum performance with sophisticated distribution mechanisms implemented through ZeroMQ messaging.

Multiple independent consumer processes can access timeslice data with individually configured policies for queuing and back pressure. The implementation treats timeslices as reference-counted work items in shared memory, allowing multiple consumers to process the same data simultaneously without duplication. A dedicated distributor process manages these references, tracking when all consumers have completed processing and automatically releasing memory when it is no longer needed.

Various queuing policies are supported: `QueueAll` guarantees delivery of every timeslice but may create back pressure, `PrebufferOne` provides opportunistic delivery to keep consumers busy but may skip items during high load, and `Skip` always waits for the newest item without queuing, optimizing for monitoring applications where latency is critical. This approach provides natural integration points for not only the optimized online data analysis, but also for quality monitoring, calibration tasks, and raw data storage while maintaining the zero-copy principle through careful buffer management.

3 Implementation and performance

The implementation of the data distribution system leverages state-of-the-art hardware and software techniques to meet the rigorous performance requirements of the CBM experiment. On the hardware side, the use of custom FPGA-based PCIe cards for data ingestion is coupled with custom DMA engines that enable zero-copy transfers directly into shared memory

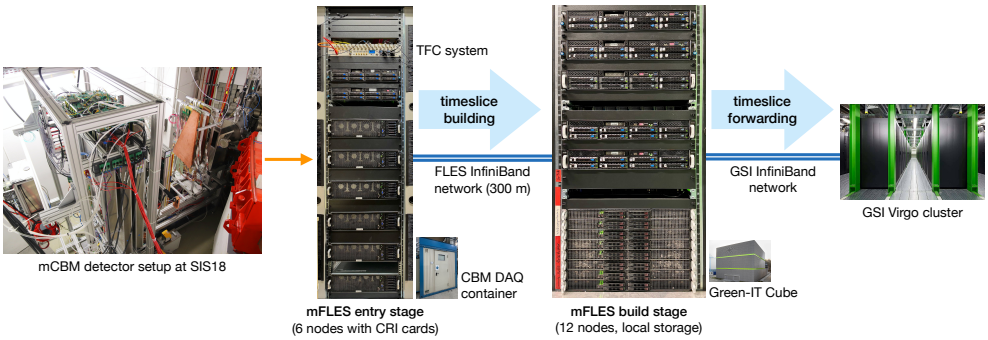

Figure 5. The mFLES system deployment at the mCBM experiment, demonstrating the integration of detector subsystems, entry nodes with custom PCIe cards (CRI), timeslice building over an InfiniBand network, and a connection to the GSI Green-IT Cube computing resources.

buffers [4]. The memory architecture is optimized to handle large blocks of data, ensuring that timeslice building operations can proceed without significant CPU overhead. The RDMA-capable InfiniBand network further minimizes latency by allowing direct memory-to-memory transfers between the entry and build nodes. See Fig. 4 for an overview of the system components.

Software optimizations focus on reducing memory access overhead and maximizing data throughput. The timeslice building application minimizes copying by aggregating complete microslice blocks into TSCs, which are then transmitted as single network transactions. In benchmark tests on a 96-node system, a sustained per-node data rate of approximately 7.2 GB/s was observed, corresponding to an aggregate system throughput of around 345 GB/s. Such performance demonstrates that the system can approach the maximum sustained bandwidth available on modern HPC platforms and that the design is able to handle the expected data rates of approximately 250 GB/s on average for the initial operation of CBM at SIS100.

The design also incorporates comprehensive error handling and flow control mechanisms. Each microslice carries a precise timestamp and error flags that enable system-wide monitoring of data acquisition consistency. These timestamps allow the system to detect when channels are falling behind or delivering data at unexpected rates. The implemented flow control mechanisms ensure that buffers are not overwhelmed by incoming data, preventing data loss and maintaining system stability.

4 System validation

The complete data distribution chain has been used successfully in several CBM-related detector tests and smaller production experiments. This includes beam tests of prototype detector setups as well as the ongoing mini-CBM (mCBM) experiment at GSI/FAIR [5].

The mCBM setup at the SIS18 contains all key components of the final system including up to 7 different detector systems including STS, TOF, RICH, TRD, BMON, MUCH, and TRD-2D (see Fig. 5). Data taking is distributed across multiple input, build, and processing nodes, connected via long-distance InfiniBand networking. Online monitoring and control interfaces have been implemented to allow for the stable operation of the data acquisition system in a production environment. This configuration enables validation of the full online processing chain under realistic conditions.

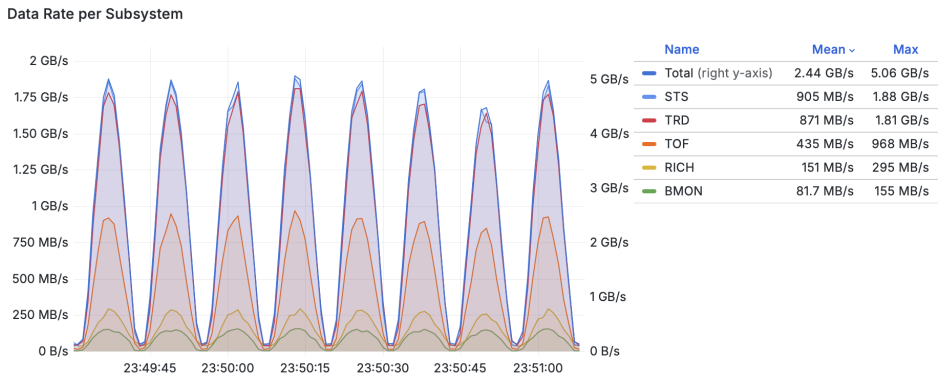

Figure 6. Example online data rates monitoring at the mCBM experiment. Shown here is the data rate per subsystem for run number 3035 on 2024-05-10.

Recent operational experience demonstrated stable performance with peak data rates above 5 GB/s (see Fig. 6) in production runs with detectors and above 10 GB/s for test pattern data, validating both the technical design and operational procedures. The system successfully handled data from 58 input channels distributed across five entry nodes and four build nodes.

5 Conclusion

The transition to a free-streaming data acquisition paradigm represents a fundamental shift in online data management for high-energy physics experiments. The approach detailed in this paper has demonstrated that a carefully engineered combination of hardware acceleration, advanced data models, and high-performance networking can meet the substantial throughput requirements of the CBM experiment. By segmenting continuous data into microslices and reassembling them into overlapping timeslices, the system ensures that events can be reconstructed efficiently and accurately online. The zero-copy data flow, enabled by RDMA and shared memory interfaces, minimizes CPU overhead and maximizes data throughput, allowing the system to approach the limits of available hardware performance. Moreover, the use of commodity HPC components, augmented by specialized front-end interfaces, provides a scalable and economically viable solution that can adapt to varying experimental conditions.

Field testing validates the design choices and implementation approach while providing valuable operational experience. The successful operation at the mCBM experiment with multiple detector subsystems confirms that the architecture can handle realistic data-taking conditions. Current development focuses on enhanced monitoring capabilities and automated configuration management in preparation for CBM commissioning. The system's modular design and scaling capabilities position it well for adaptation to evolving experiment requirements.

This work is supported by BMBF (05P21RFFC1).

References

[1] T. Ablyazimov, A. Abuhoza, R.P. Adak et al., Challenges in QCD matter physics – The scientific programme of the Compressed Baryonic Matter experiment at FAIR, Eur. Phys. J. A **53**, 60 (2017). 10.1140/epja/i2017-12248-y

[2] J. de Cuveland, D. Emschermann, V. Friese, I. Fröhlich, P. Gasik, D. Hutter, W. Müller, C. Sturm, eds., Technical Design Report for the CBM Online Systems – Part I, DAQ and FLES Entry Stage, CBM Technical Design Reports (GSI, Darmstadt, 2023), https://doi.org/10.15120/GSI-2023-00739

[3] J. de Cuveland, D. Hutter et al., CBM FLES timeslice building, Git repository, https://github.com/cbm-fles/flesnet

[4] D. Hutter, An input interface for the CBM first-level event selector. Dissertation (Goethe University, Frankfurt am Main, 2021), https://nbn-resolving.org/urn:nbn:de:hebis:30:3-591599

[5] CBM Collaboration, mCBM@SIS18 – A CBM full system test-setup for high-rate nucleus-nucleus collisions at GSI / FAIR (GSI, Darmstadt, 2017), https://doi.org/10.15120/GSI-2019-00977