PROCEEDINGS OF SCIENCE

Triggerless data acquisition system for the AMBER experiment

Martin Zemko,[a,c,*] **Dominik Ecker,**[f] **Vladimir Frolov, Stefan Huber,**[f] **Vladimír Jarý,**[c] **Igor Konorov,**[f] **Antonín Květoň,**[b] **Josef Nový,**[c] **Benjamin Moritz Veit**[a,d] **and Miroslav Virius**[c]

[a] *CERN, Geneva, Switzerland*
[b] *Charles University in Prague, Prague, Czech Republic*
[c] *Czech Technical University in Prague, Prague, Czech Republic*
[d] *Johannes Gutenberg University, Mainz, Germany*
[f] *Technical University of Munich, Munich, Germany*

E-mail: martin.zemko@cern.ch

We developed a novel free-running data acquisition system for the AMBER experiment. The system features a hybrid architecture containing a scalable FPGA-based system for data collection and conventional distributed computing for data reduction. The current implementation can collect up to 10 GB/s sustained data rate. The FPGA system substitutes high-performance networks by merging time-correlated data and distribution between computers. The data reduction is performed by a filtering farm decreasing the incoming data rate by a factor of 50 to 100-200 MB/s. The filtering framework implements various data reduction algorithms for different physics programs. These algorithms perform partial data decoding, time, and spatial analysis of the data in order to select predefined event topology in a semi-online manner. Our system also performs continuous and iterative time calibration of detectors, which is required by the continuously running acquisition system. Additionally, we developed a simulation tool able to emulate detector responses to particles passing the AMBER spectrometer and convert them into correctly formatted raw data. These generated data are used to test and validate the readout chain and the filtering framework. The entire system will be tested with a limited number of detectors this year. The first physics run is planned for 2024.

41st International Conference on High Energy physics - ICHEP2022
6-13 July, 2022
Bologna, Italy

*Speaker

PoS(ICHEP2022)248

1. AMBER experiment

The AMBER experiment is a fixed target experiment located at the M2 beamline of the Super Proton Synchrotron at CERN. It was approved by the CERN research board in 2021 and took its first pilot run this year. The M2 beamline can provide high-intensity hadron or muon beams suited for various physics goals. The AMBER experiment includes several physics programs starting with a proton radius measurement using elastic muon-proton scattering. This program requires fewer detectors with a small data rate. Later on, Drell-Yann pair production using conventional hadron beams will follow. This measurement includes a higher number of detectors, and the overall data rate will also be increased. [1]

The most challenging physics program, in terms of data acquisition, is a measurement of the proton radius by elastic muon-proton scattering using the muon beam impinging on a hydrogen time projection chamber (TPC). The high-pressure TPC filled with hydrogen (reaching up to 20 bars) will measure the energy of recoil protons. The TPC will be surrounded by four unified tracking stations consisting of scintillating fibers (providing timing) and silicon pixel trackers (providing positioning). These stations will be able to measure the angle of scattered muons with the precision of 5 µm [2]. The hydrogen TPC has a very long drift time (approximately 120 µs), which is in contrast to other detectors with a response time below 500 ns. Due to a low probability of recoil proton detection, the amount of data can be reduced approximately by factor 50. To merge the data from all these detectors, a new triggering paradigm needs to be exploited. [1]

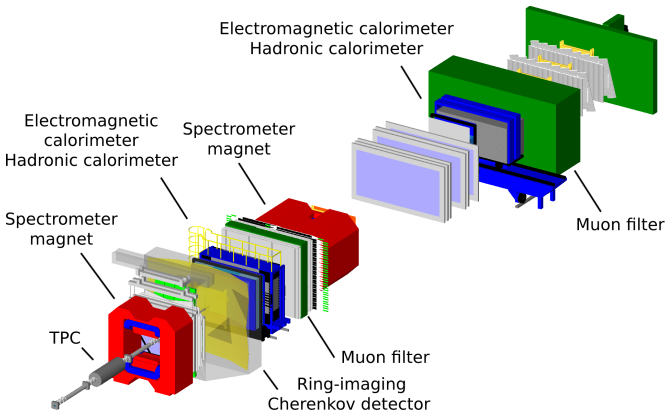

Figure 1: Isometric view on the AMBER experimental setup [1]

2. Data acquisition system

AMBER will use a streaming, triggerless acquisition system based on continuous detector readout. As a replacement for low-level trigger logic, the system utilizes a high-level filter running in a semi-online mode that gives unrestricted time to make a decision. Moreover, the system based on traditional computing is flexible enough to allow any detector to participate in the filtering process. According to our initial estimation, the overall sustainable data throughput of the readout system shall be at the level of 10 GB/s. [3] Such data flux puts significant requirements on available resources and equipment. To reduce the cost and size of the system, many optimizations are needed at all levels.

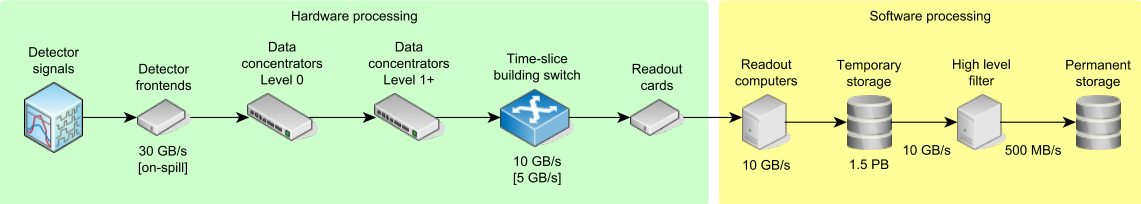

Figure 2: Simplified structure of the streaming acquisition system [4]

2.1 Data protocol

Firstly, we developed a data structure for continuous data taking. All data are time tagged and organized in blocks. The smallest data unit is a so-called image containing information from a single detector within a given period that depends on the response time of the detector. It is worth noticing that smaller images cause higher protocol overhead, whereas large images result in high data fluctuations and inefficient data reduction. Therefore, it is a trade-off between image size and stable data rate.

Further, images are organized in time slices containing data from all detectors within a specified time interval. Time slices are time-based equivalents of events and independent elementary processing units that can be processed in parallel. The slice duration is a parameter of the acquisition system that needs to be optimized according to the final detector setup. Similar to images, small slices cause high fluctuations in data rates and protocol overhead. [4]

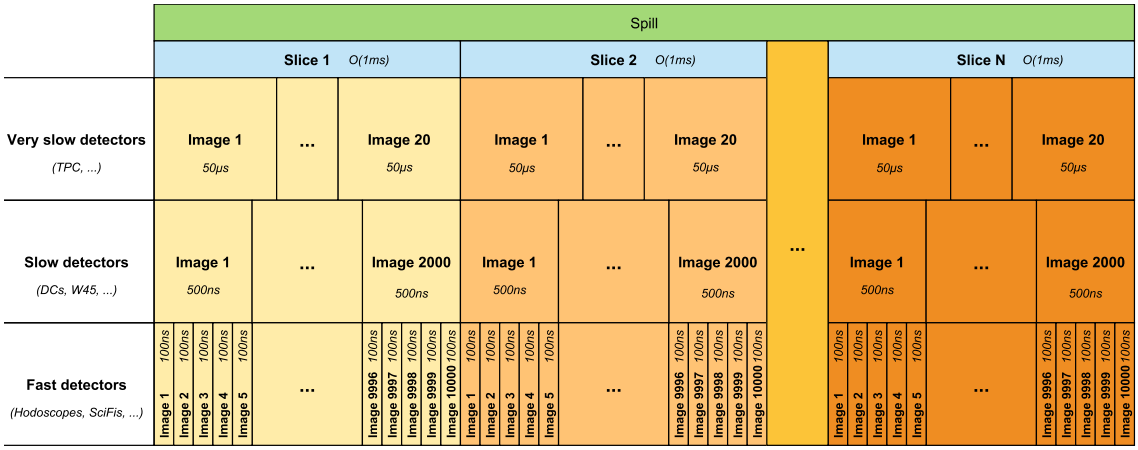

Figure 3: Data structure of the streaming acquisition system [1]

2.2 Readout software

The AMBER readout software is a distributed system based on master-slave architecture. The master process manages and controls all slave processes that perform readout. It also monitors the states of all modules and keeps the system in a consistent state thanks to remote monitoring of its child processes. If one slave process fails, the master automatically restarts it and performs an automatic error recovery procedure. As a consequence, the readout system handles unstable conditions. [4]

Data transmission from the PCIe driver to the local storage is done by slave processes running on readout engines. Each slave controls a single server and the attached readout card. All processes support multi-threading, and their tasks are distributed to worker threads. The number of threads varies according to available resources on the server. Moreover, threads respect associations of data with certain NUMA domains (CPUs), i.e., instead of moving data between NUMA domains, slices are processed on the domain where they are stored. This approach minimizes the utilization of processor interconnection and improves performance.

3. High level filter

The described readout system is optimized for maximal throughput and can produce up to 1 GB/s of data per compute node. Naturally, not all data is useful for physics analysis; most data represents uninteresting events. Therefore, such information is removed by the high-level filter (also known as HLT). The HLT is a distributed computational framework that works in asynchronous mode and is fully detached from the online DAQ system. Similarly to the main readout system, the HLT is optimized for an arbitrary number of NUMA domains. Hence, it can efficiently exploit all CPUs within a single server. The system automatically utilizes all available resources and distributes time slices to all threads. Thanks to the independent nature of slices, they are executed on many processors in parallel.

At first, data are transmitted from the local storage via a 25 Gbps network to filtering nodes using the RDMA (remote DMA) interface that provides fast data transfers between computers. Then, raw data files are read, and individual time slices are identified within the raw data stream. Each slice is parsed to the level of images, and its object representation is built in memory. In addition, images produced by trigger detectors are decoded in order to extract trigger information from their data words. Such information comes in various forms (e.g., hit times, positions, amplitudes, etc.), and it is forwarded to a filter pipeline for further processing. At the same time, other images are kept in the buffers and wait for the final image filtering. The filter pipeline consists of two main stages: [4]

1. **time analysis** (preprocessing) – only hit times are taken into account. Timing algorithms search for hit clusters describing particle interactions (event candidates) in the time domain. Meantime is calculated for each cluster and sent to the next stage for spatial validation.

2. **spatial analysis** (filtering) – only hit positions are considered. Individual event candidates are examined, and the topology of associated hits is evaluated. If the topology meets the conditions defined in the filter algorithm, the event candidate is marked as valid, and all time-correlated images are tagged for saving. For each event candidate, we always mark two consecutive images to avoid edge cases and loss of information.

Finally, all marked images are written to the output file alongside the list of valid event candidates. While all other images are dropped. Thus, the final amount of data is reduced without changing the data structure. The efficiency of data reduction depends on the selected algorithm and its parameters.

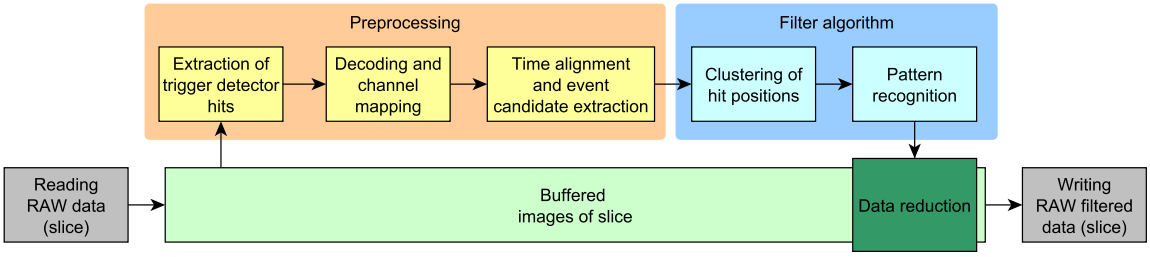

Figure 4: Processing pipeline of a single filtering thread [4]

4. Calibration and alignment

In triggered systems, trigger detectors must be time-aligned and calibrated in advance. In other words, hits from one trigger detector must coincide with hits from another; only then the detectors can be used for event identification. In the new streaming system, the HLT requires the timing and relative positioning of individual detectors in advance. Since combinations of detectors included in filter algorithms are more diverse, the complexity of calibration and alignment is higher. To gain some extra time to calculate the coefficients, we take advantage of large buffers (disks) attached to readout engines that provide a sufficient capacity to store several days of data-taking. This period is used to calculate coefficients and load fine time corrections back to front-end cards.

This manual method shall be used only during the commissioning phase. Later, an automatic system of calibrations will be deployed. Initially, detector experts perform a rough calibration with a precision better than a single image length so that hits will be within the analysis time window. Once this coarse calibration is done, the second phase takes place. During this phase, the system performs the calibration automatically. At first, a certain amount of statistics is collected; then, the HLT calculates residuals from the event time (T_0 interaction time) for each detector separately. If these residuals exceed a certain threshold, coefficients are compensated for the difference, and new values are loaded to front-end cards. Afterward, this process repeats, and the sum of residuals approaches zero after several iterations. This iterative method creates a feedback loop that calibrates detectors in the time domain.

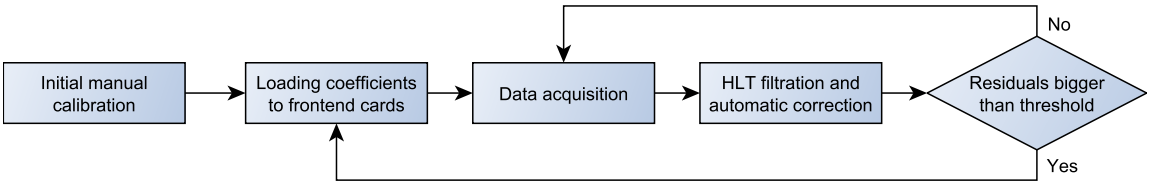

Figure 5: Algorithm of automatic time calibration of detectors

5. DAQ emulation tool

Due to the necessity of DAQ development in parallel with detector production, we could not test the system with real detector data. But still, we needed a way to continue with HLT development even without the final detector setup. Therefore, we designed a tool capable of full emulation of the acquisition system. At the input, we provide physics events generated by Monte Carlo simulations.

This information is provided in an event-based format. Afterwards, it is converted into a streamlined data format.

As a part of this process, we simulate responses of individual detectors to passing particles (detector response simulation) and responses of frontend electronics (readout simulation). In addition, we apply several other corrections, such as time-of-flight correction, time shift, etc. At the end of the simulation chain, we obtain a full-sized data stream as it would be produced by the DAQ system. Then, output data are directly submitted to the HLT system and processed. Presumably, individual events are still fully recognizable by filtering algorithms. Since the HLT can reconstruct events and filter out the data, we can closely watch the filtering process, validate, and debug the filtering algorithms. Additionally, thanks to links to original Monte Carlo data, we can compare the outcome of the filtering procedure with the expected results.

6. Conclusion

AMBER is an upcoming experiment with challenging physics programs that require a new triggering approach. Therefore, we developed the streaming acquisition system based on the continuous readout. Individual detectors send as much data as possible using the custom streaming protocol. Its data format is optimized for this use case and allows smooth transmission of various data words with different frequencies. Additional headers are being removed by readout servers performing several data checks and validations. The final data treatment and filtering happen in the high-level filtering farm optimized for high throughput (1 GB/s per node) and fast data reduction. It also supports various filtering algorithms such as matrix, random, or tracking filters. Besides, we designed the iterative system of calibrations, which provides continuous updates of calibration values. To validate readout processes and filtering algorithms, we developed the emulation tool to convert Monte Carlo physics data into streamlined data containing fully reconstructable events. This emulation tool is used to evaluate and debug the filtering system prior integration of real detectors in DAQ.

References

[1] B. Adams, C. A. Aidala, G.D. Alexeev, et al., *Proposal for Measurements at the M2 beam line of the CERN SPS*, 2019.

[2] AMBER collaboration, *AMBER Status Report 2022*, CERN-SPSC-2022-023, 2022. https://cds.cern.ch/record/2810822/files/SPSC-SR-313.pdf

[3] S. Huber, I. Konorov, A. Kveton, et al., *Data Acquisition System for the COMPASS ++ / AMBER Experiment*, 2020.

[4] M. Zemko, V. Frolov, S. Huber, et al., *Free-running data acquisition system for the AMBER experiment*, 04028, 1–12, 2021.

[5] Y. Bai, M. Bodlak, V. Frolov, et al., *Overview and future developments of the FPGA-based DAQ of COMPASS*, Journal of Instrumentation, vol. 11, no. 02, p. C02025, 2016.