Triggerless data acquisition system for the AMBER experiment

V. Frolov, S. Huber, V. Jarý, I. Konorov, A. Květoň, D. Levit,
J. Nový, D. Steffen, B. M. Veit, M. Virius, **Martin Zemko**

AMBER DAQ group and CERN

ICHEP 2022, 8th July 2022

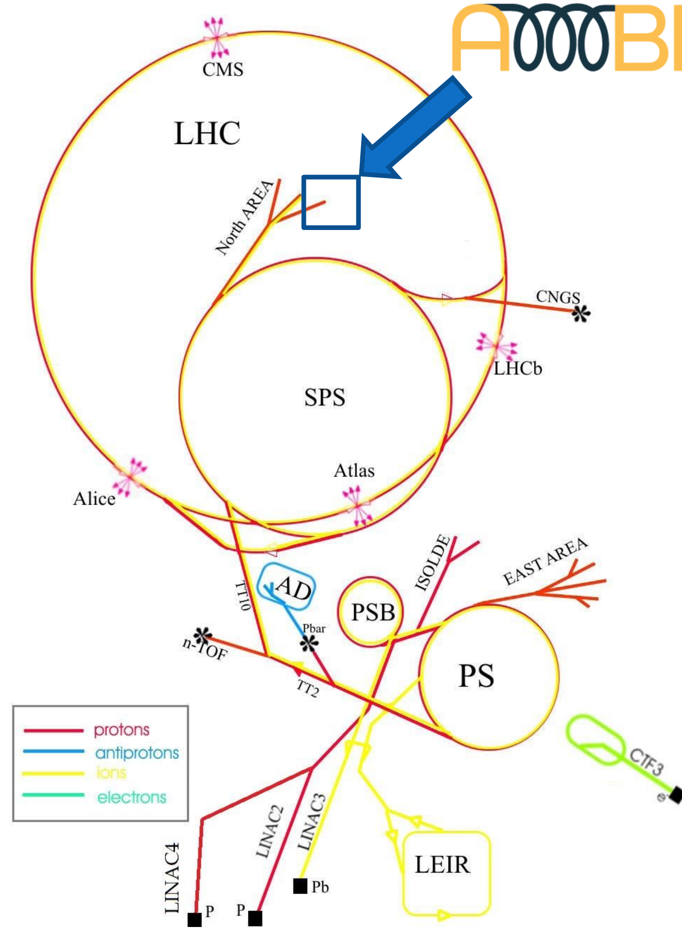
CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

CERN

AMBER
Apparatus for Meson and Baryon Experimental Research

AMBER Experiment

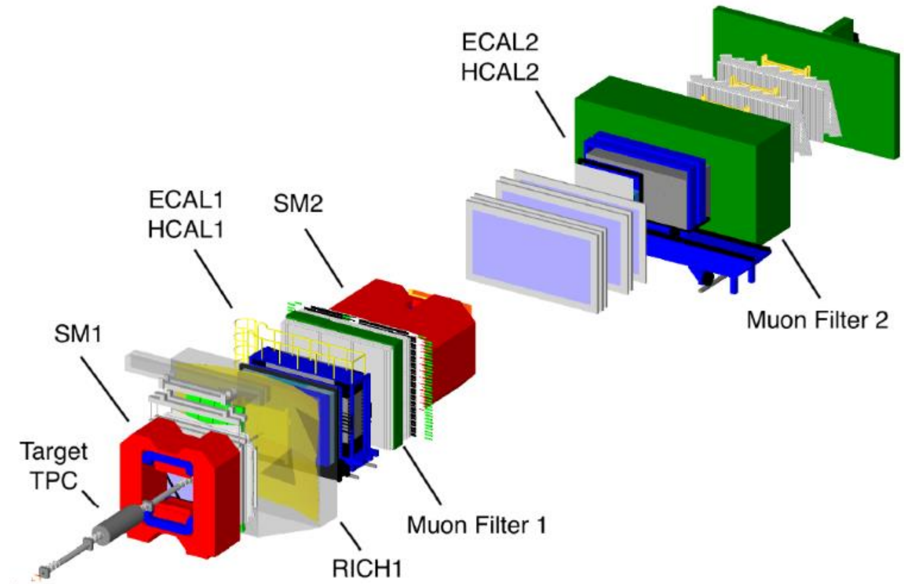
- AMBER is a fixed target experiment located at the M2 beam line of the CERN SPS
- It is the successor of the COMPASS experiment
- It has been approved by the CERN research board in 2021
- Its first pilot run took place in 2021
- Next test run will follow this year
- First main objective of the free-running DAQ is to **measure the proton radius** on an active hydrogen time projection chamber (TPC) with a muon beam



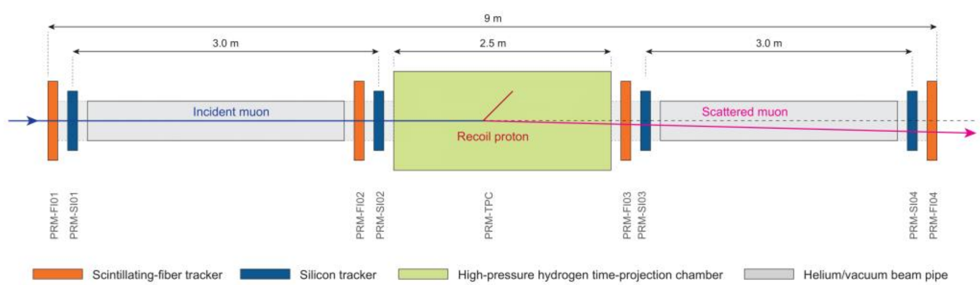
CERN accelerator complex

Physics programmes and goals

- High pressure time projection chamber (TPC) filled with hydrogen (up to 20 bars) is used as an active target
- Muon scattering angle calculated from four tracking detectors (fast detectors)
- Slow detectors (e.g. hydrogen TPC) have very long drift time (120 μs)
- Merging slow detectors with fast detectors → streaming DAQ

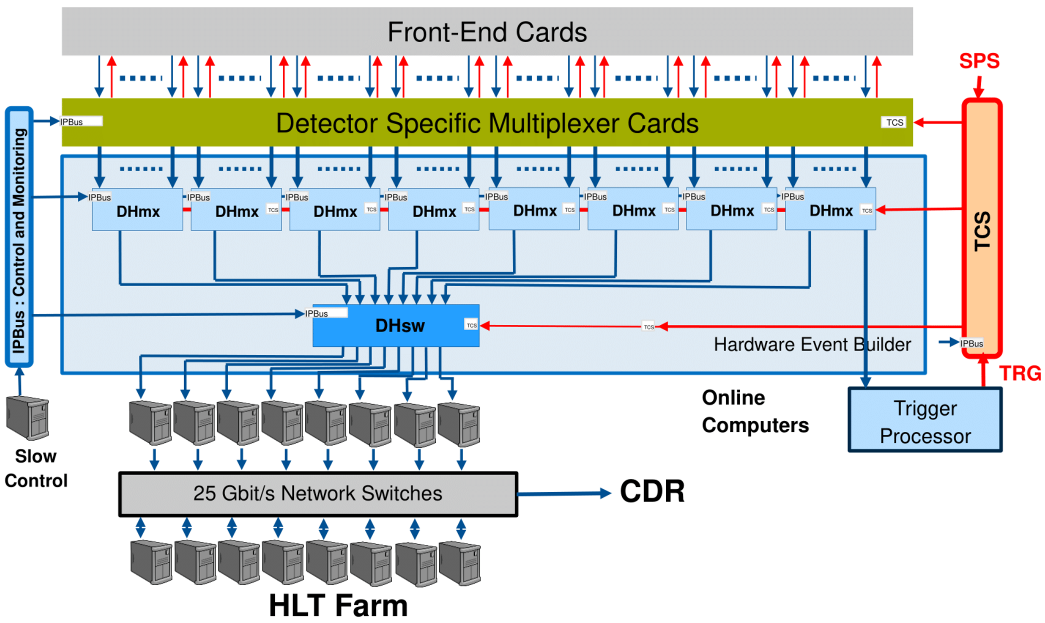


Isometric view on the AMBER experiment



Central part of the detector setup

Triggerless data acquisition

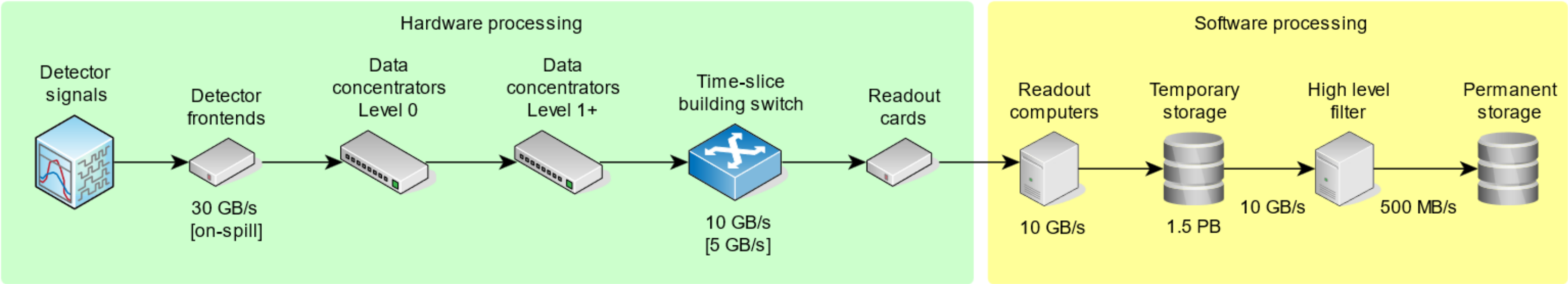


Structured scheme of the acquisition system

- AMBER uses a **triggerless data acquisition system** → continuous readout of detectors
- High level filter (HLT) is used instead of a low-level trigger logic → more time for the data reduction
- **General reduction scheme** → any detector can participate in the filter
- To achieve our **goal of 10 GB/s**, we need high performance software and hardware

Readout chain

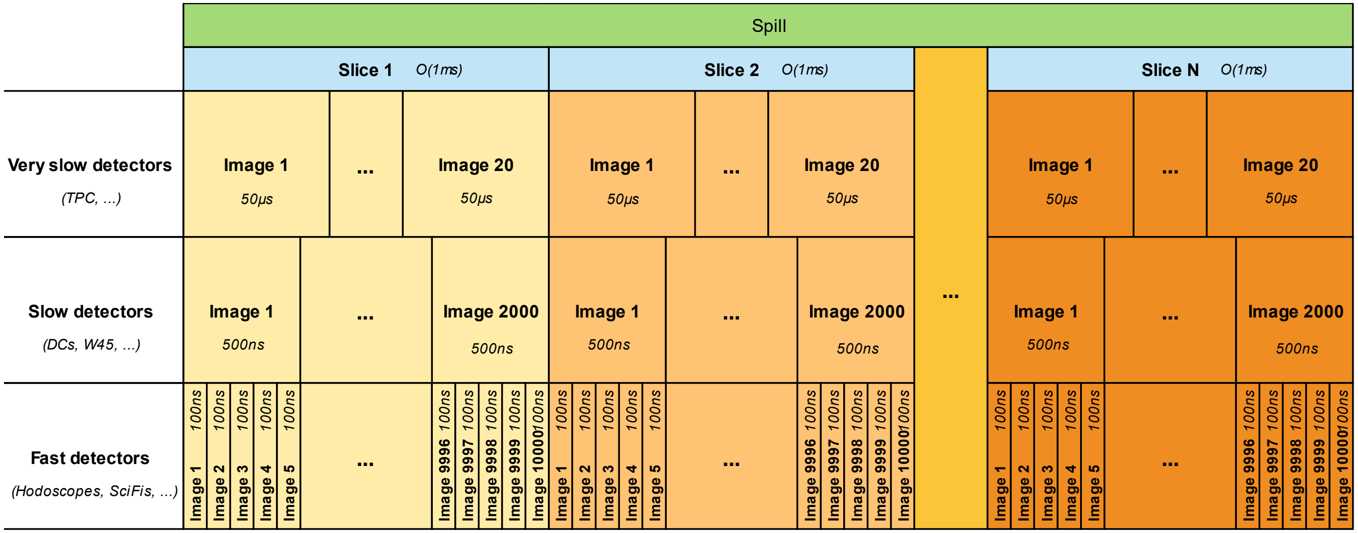
- At the first stages, FPGA hardware modules process the data and perform multiplexing and time-slice building (high uptime of 98%)
- Subsequently, data are processed by readout cards accompanied by readout software
- Filtering software (HLT) performs the final data filtration
- There are several parallel streams at each level → high scalability



Triggerless acquisition system (AMBER)

Data structure

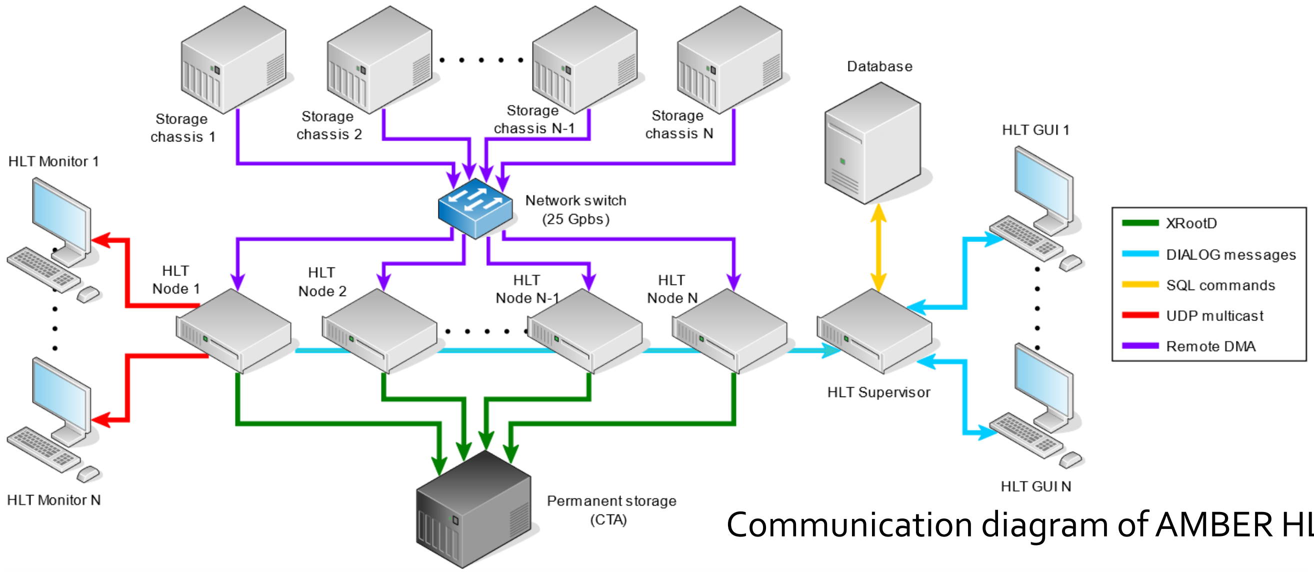
- We developed a custom data protocol consisting of several layers of encapsulated data
- At the lowest level, data are divided into **images** → **data from a single detector** within a given interval (depends on detector time response)
- Images are incorporated into **time slices** → **data from all detectors** within a specified time interval, time-based equivalent to events, independent elementary processing units
- Slice duration is a parameter for optimization O(1ms) → trade-off between time slice size and data size fluctuation (stable data rate)



Triggerless data protocol

HLT framework

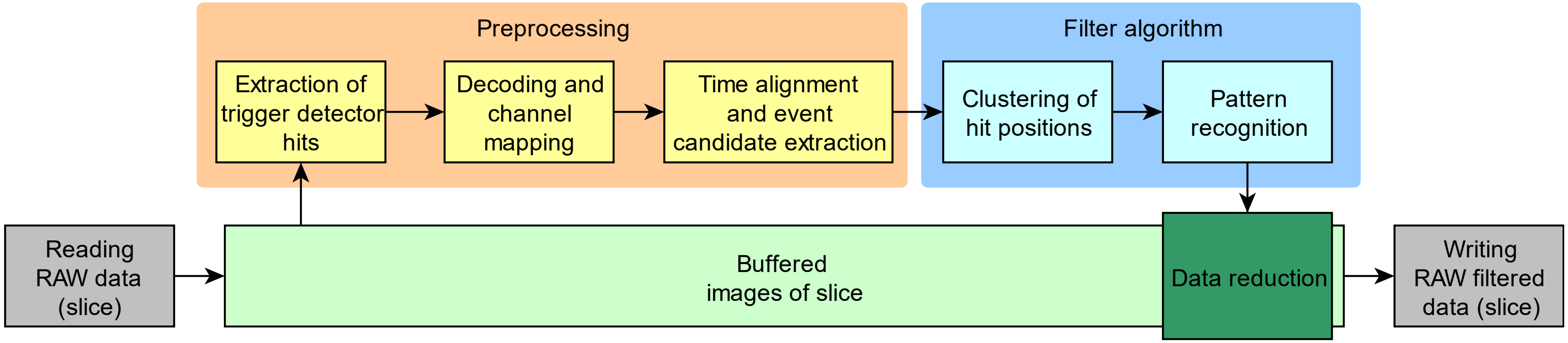
- HLT is a high-performance distributed computational framework based on a master-slave architecture
- Slices are processed on many nodes and threads in parallel
- Written in C++ and and the Qt framework
- Provides advanced libraries for inter-process communication, networking, databases, filter algorithms, etc.
- Algorithms are modular and highly optimized (possibility to implement new filter algorithms)



Communication diagram of AMBER HLT

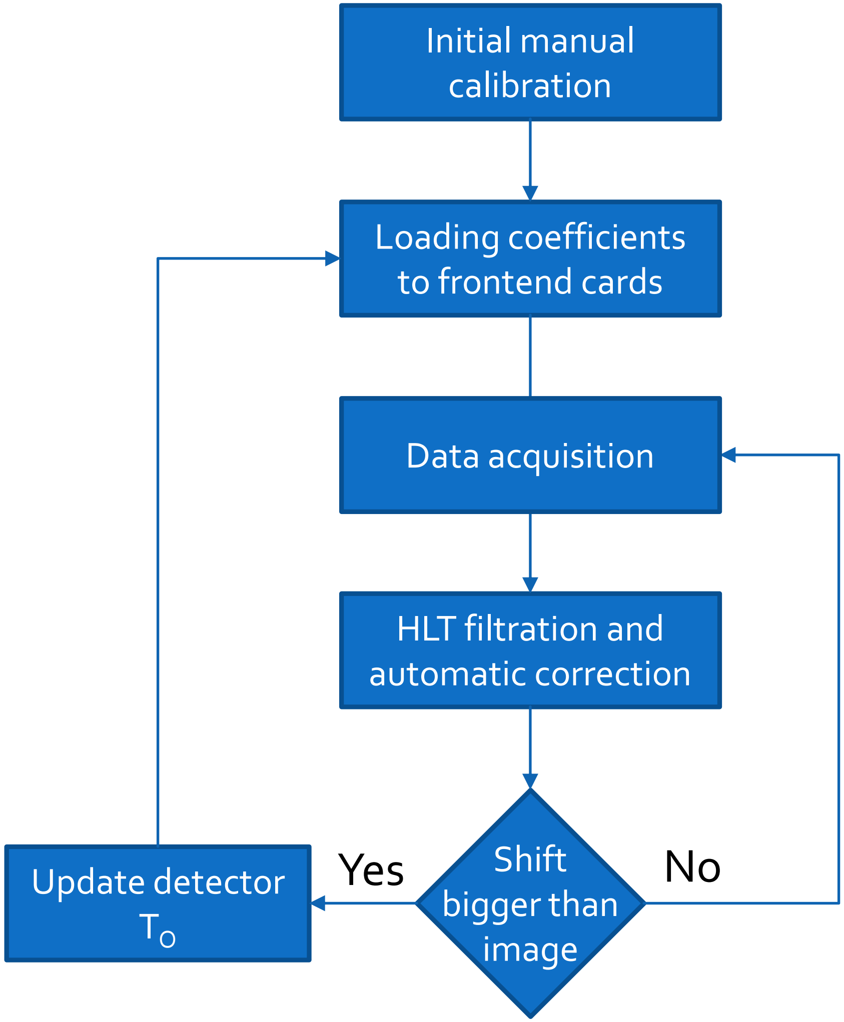
Filtering steps

- Filtering application is the heart of the HLT framework
- Each thread / node repeats the same sequence of steps – filtering algorithm:
  1. Extraction of detector information for filter decision
  2. Data used for filtration are decoded and analysed in the time dimension
  3. Afterwards, their spatial properties are taken into account and a binary decision is made – keep the data or drop them
  4. Finally, the decision is applied on raw data – data size is reduced
  5. Data are written to an output file



HLT Filtration pipeline

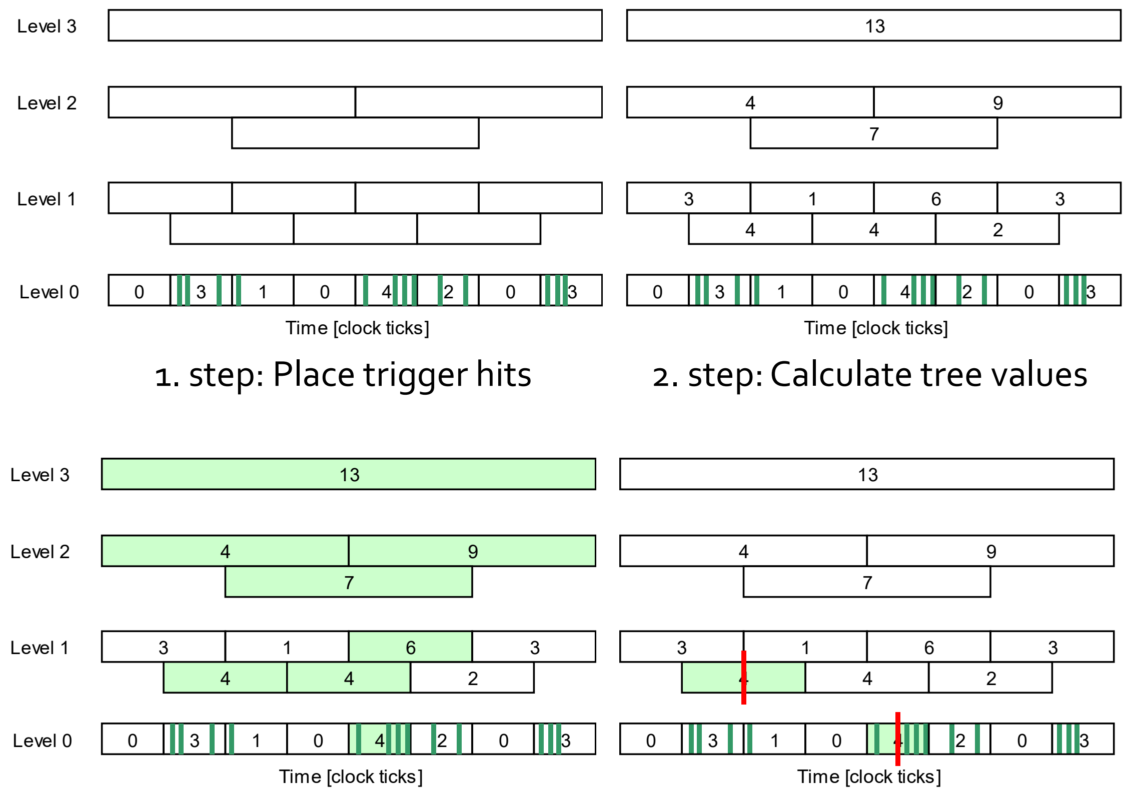
Continuous time calibration



- HLT filter processes and decodes „trigger hits"→ they must be aligned in time
- HLT compensates for time drifts within a single image (see the next slide)
- If hits are shifted more than a single image width, recalibration is needed
- Newly calculated coefficients are loaded back to frontend cards

Identification of event candidates

- For the timing analysis, we use a modified version of the Haar wavelet decomposition – **shifted wavelet decomposition**
- Sum of adjacent data fields is a sufficient aggregator for event identification
- Our implementation uses 8 levels to achieve the desired resolution and time range
- Output of the SWT algorithm is a list of so-called **event candidates**



1. step: Place trigger hits

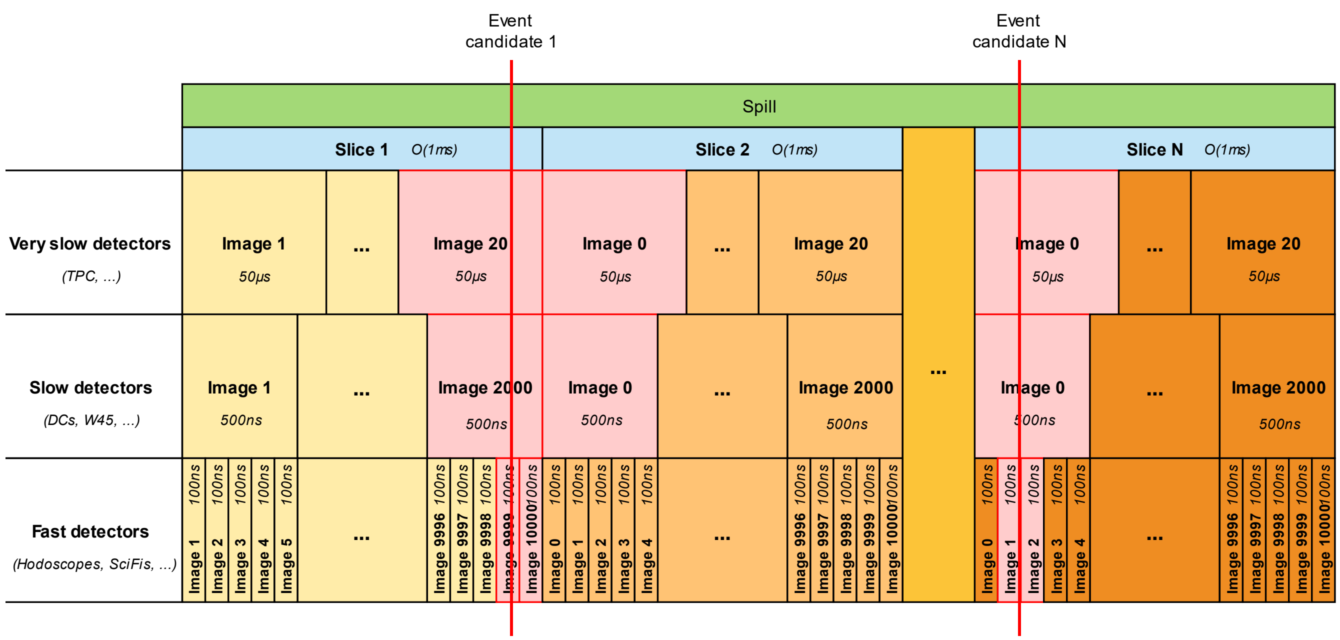
2. step: Calculate tree values

3. step: Activate cells above certain threshold (4)

4. step: Prune parent cells

Data reduction

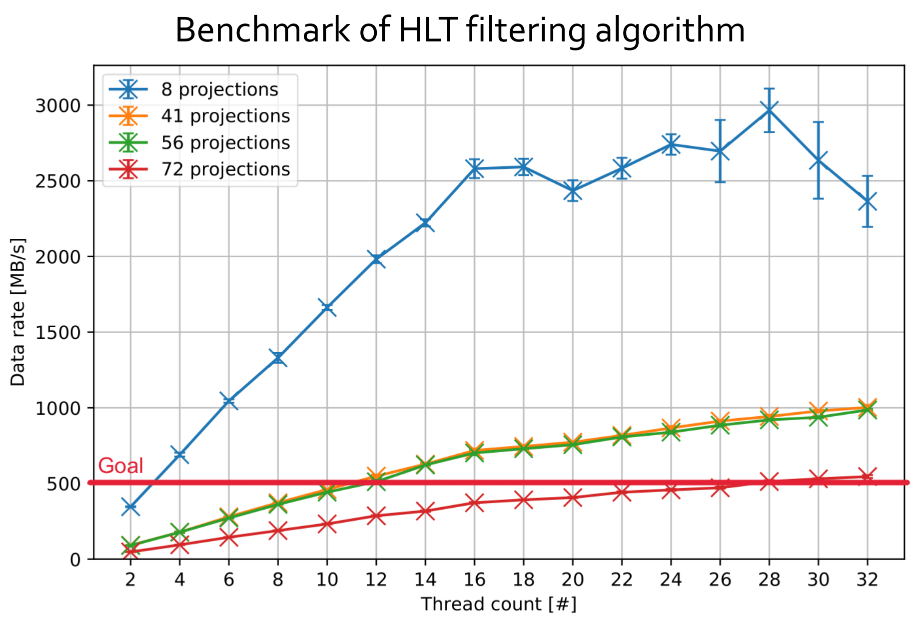
- If an event candidate is identified in the data, two consecutive images of each detector are saved
- Last image of a slice is also copied to the next slice in order to prevent edge cases
- All images that are not associated with any event candidates are removed
- Eventually, HLT appends a list of valid event candidates at the end of slice



Priciple of reducing data size

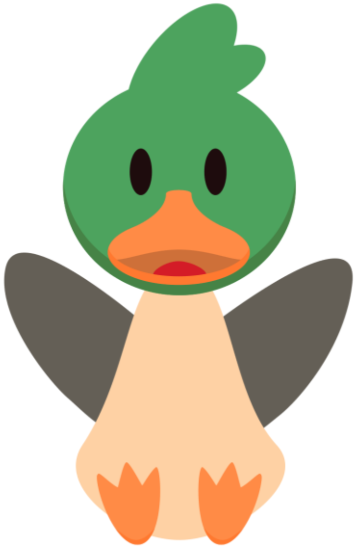
HLT benchmarks

- We used the emulation tool to test, validate, and measure HLT performance
- HLT filter was tested with artificial data in a limited scope including two servers and simple filtration algorithms
- Test setup:
  - AMD EPYC (16 cores, 32 threads)
  - 128GB DDR4 memory
- Processing rate depends on:
  - number of threads,
  - used filter algorithm,
  - number of projections (planes),
  - slice duration, etc.



Summary

- We developed fundamental components for the triggerless DAQ system of the new AMBER experiment at CERN
- System introduced the custom data protocol and the high-level filtering framework replacing the low-level trigger
- We created the HLT framework
  - We developed methods for continuous time calibration of detectors
  - Performance of the high level trigger has been evaluated and measured
- Reduced system will be tested in the upcoming test run at the end of 2022

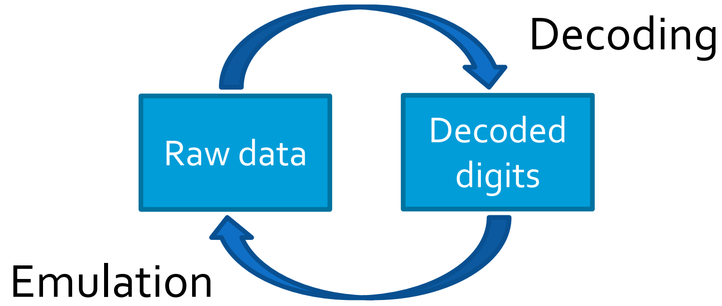


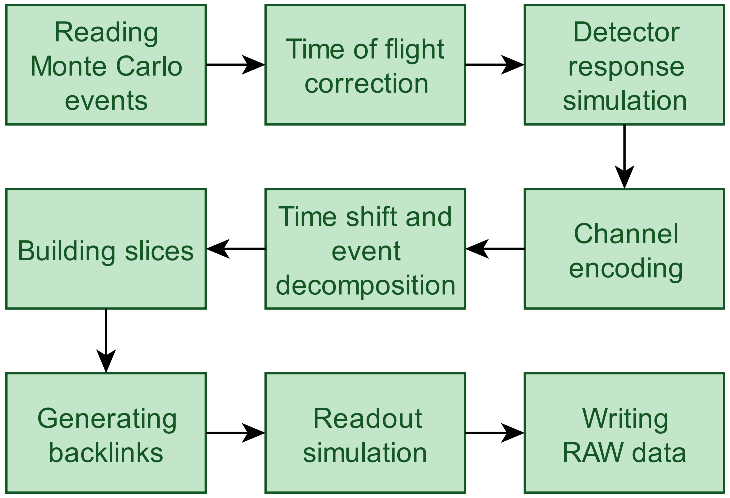
Thank you for your attention

Spares

Simulation chain

- We needed some ways of testing the HLT software and verification of its functionality
- We developed a DAQ emulation tool called **Data Generator**
- It transforms **Monte Carlo events** into **raw data**
- Data Generator parses Monte Carlo events, simulates detector and frontend responses, and produces data stream in the new DAQ format
- At the output, we also obtain links to original Monte Carlo events used for validation





Individual steps of the simulation chain