# Fast online reconstruction and online calibration in the ALICE High Level Trigger

David Rohr for the ALICE Collaboration,
drohr@cern.ch
Frankfurt Institute for Advanced Studies

Real Time 2016, Padua, 10.6.2016

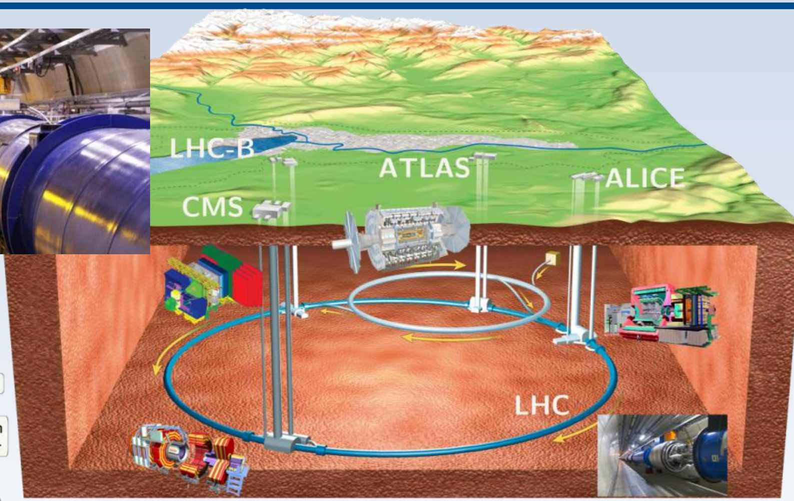
BMBF Forschungsschwerpunkt ALICE Experiment 202

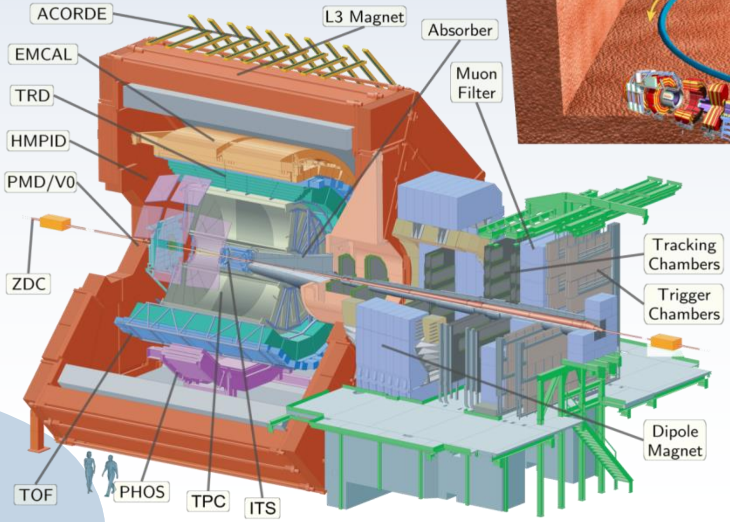
ALICE

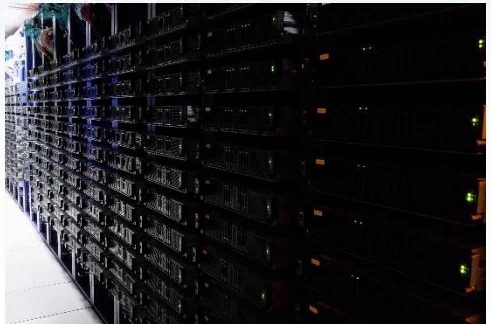
ALICE

# ALICE at the LHC

- The Large Hadron Collider (LHC) at CERN is today‘s most powerful particle accelerator colliding protons and lead ions.
- ALICE is one of the four major experiments, designed primarily for heavy ion studies.
- The Time Projection Chamber (TPC) is ALICE’ primary detector for track reconstruction.
- The High Level trigger (HLT) is an online compute farm for real-time data reconstruction for ALICE.





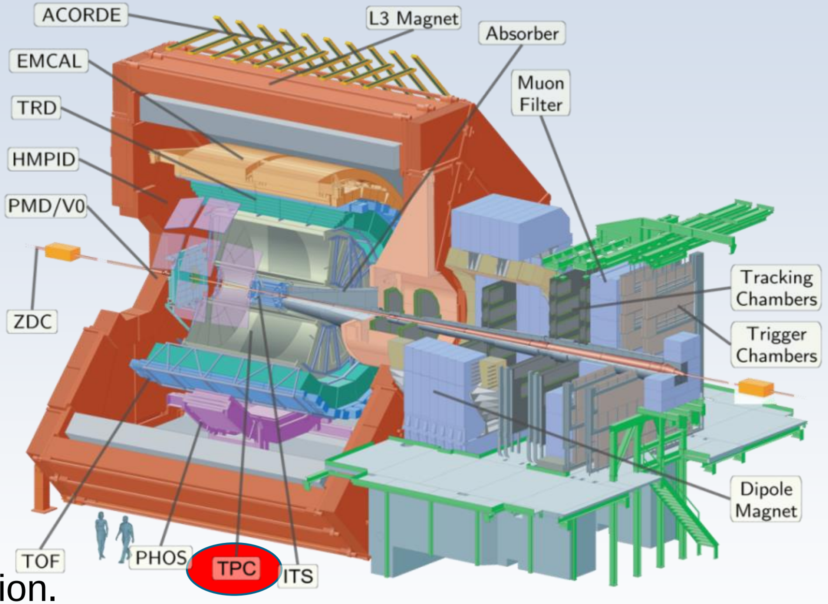


# Why online calibration in the HLT?

- **Reconstruction of particle trajectories in the TPC is computationally very expensive:**
  - Several thousand tracks per event.
  - High combinatorial complexity.

- **As a gas-based detector, the TPC is sensitive to calibration.**
  - Environment variables such as temperature and pressure affect the calibration.
  - The conditions change during a run.

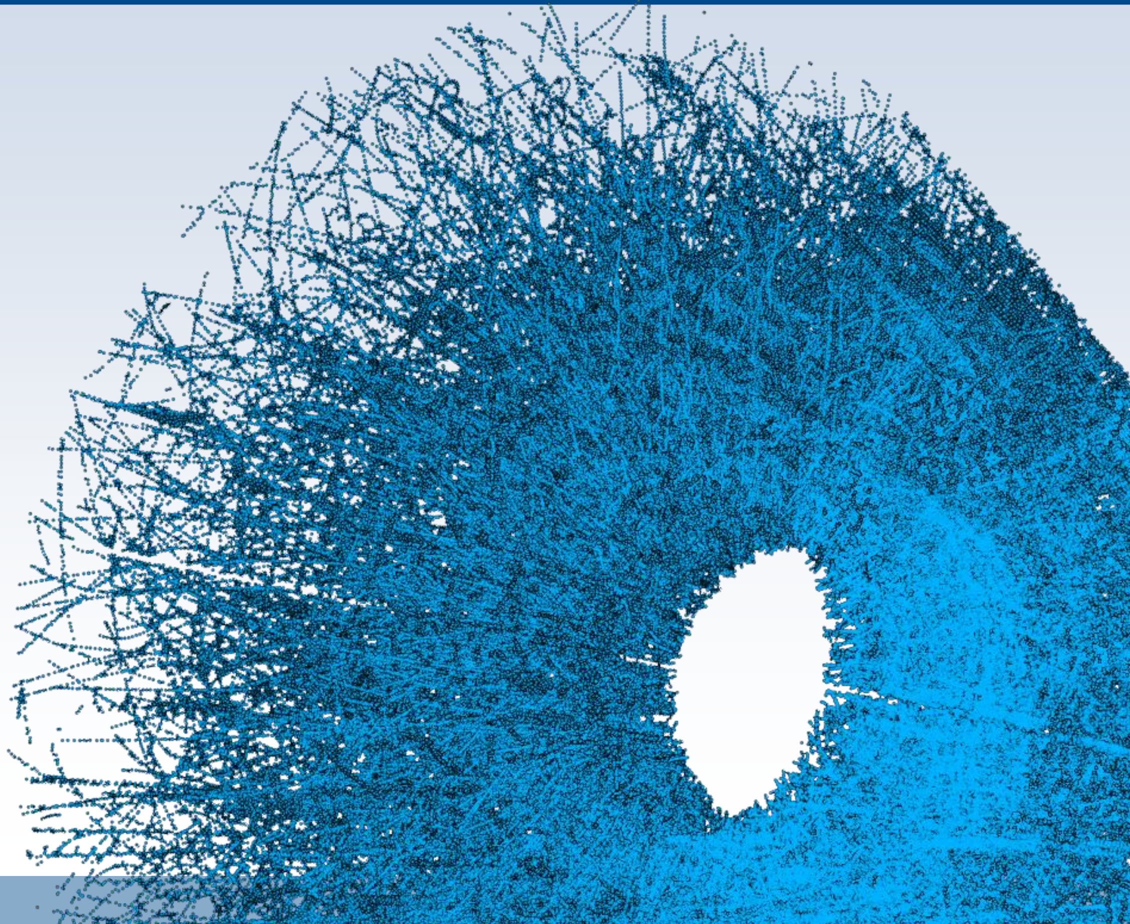
→ **Challenging tasks for the HLT:**
  - Needs fast reconstruction algorithm for online operation.
  - Detectors must be continuously calibrated online.



- **Online calibration improves online reconstruction quality.**
- **Online calibration can save offline compute resources by replacing offline calibration passes.**
- **Future experiments (ALICE in Run 3, FAIR at GSI) rely on online processing and thus online calibration.**

# Tracking at ALICE

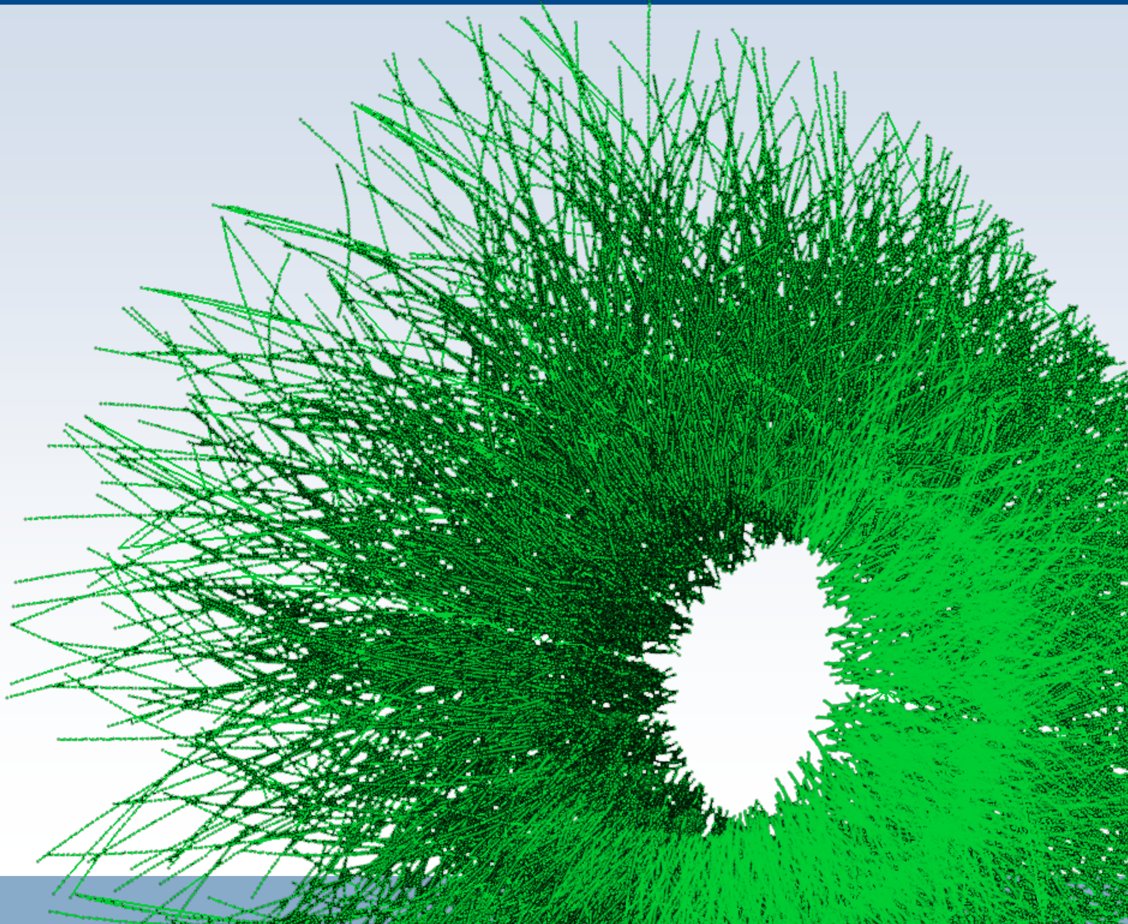
## ALICE events:



TPC Hits in heavy ion collision.
(Set of 3d space points)

# Tracking at ALICE

## ALICE events:



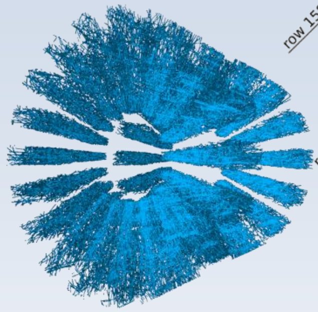
TPC tracks in heavy ion collision.
(Trajectories connecting the points)

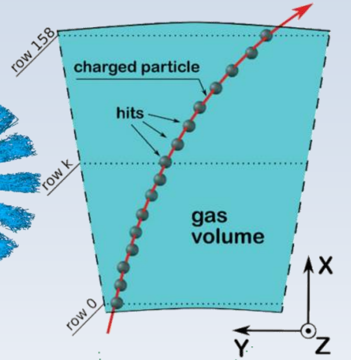
# Baseline from Run 1: Fast TPC GPU Tracking

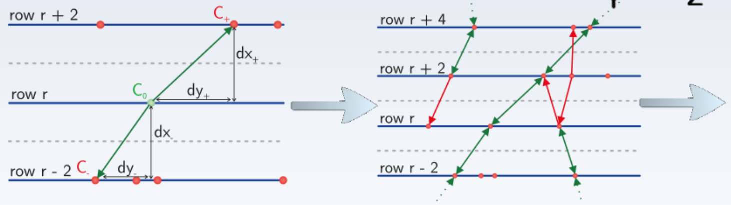
- **TPC Volume is split in 36 sectors.**
  - The tracker processes each sector individually.
  - Increases data locality, reduce network bandwidth, but limits parallelism.
  - Each sector has 160 read out rows in radial direction.

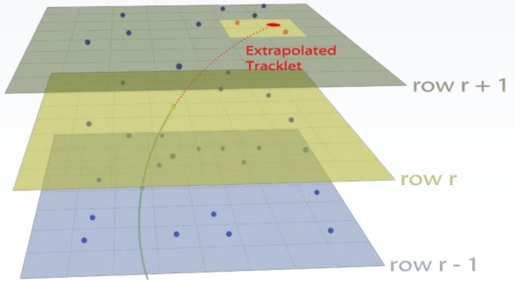
- **1. Phase: Sector-Tracking (within a sector)**
  - Heuristic, combinatorial search for track seeds using a **Cellular Automaton**.
    - A) Looks for three hits composing a straight line (**link**).
    - B) Concatenates links.
  - Fit of track parameters, extrapolation of track, and search for additional clusters using the **Kalman Filter**.

- **2. Phase: Track-Merger**
  - Combines the track segments found in the individual sectors.
  - Performs the final track fit.









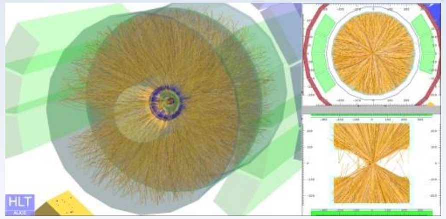
# Summary of HLT TPC Tracking

- **HLT Tracking 15 times faster than offline tracking.**
- **GPUs speed up HLT tracking by additional factor of 10 → Total speedup 150.**

- **GPU and CPU results consistent and reproducible.**

- **GPU Tracker runs on CUDA, OpenCL, OpenMP – one common shared source code.**

- **Now: 180 compute nodes with GPUs in the HLT**
  - First deployment: 2010 – 64 GPUs in LHC Run 1.
  - Since 2012 in 24/7 operation, no problems yet.



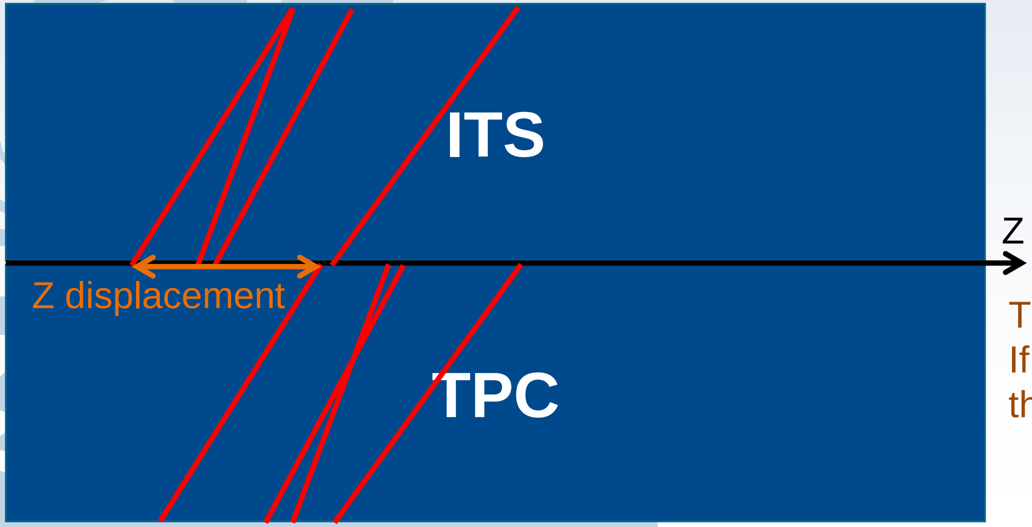
- **Cost savings compared to an approach with traditional CPUs (HLT CPU tracker):**
  - About 500.000 US dollar during ALICE Run I.
  - Above 1.000.000 US dollar during Run II.

  - Mandatory for future experiments, e.g.. CBM (FAIR, GSI) and ALICE $O^2$ with >1TB/s data rate.
- **See e.g. CDT 2015** (https://indico.physics.lbl.gov/indico/contributionDisplay.py?contribId=17&sessionId=8&confId=149)

## Why online calibration in the HLT?

- Our tracking can only be as good as our clusters.
  - For online computing, we often ignore calibration, or we take just „some“ calibration.
  - This can be sufficient for fast online processing in some cases (e.g. track finding), if the effect is constant, linear, or at least smooth.

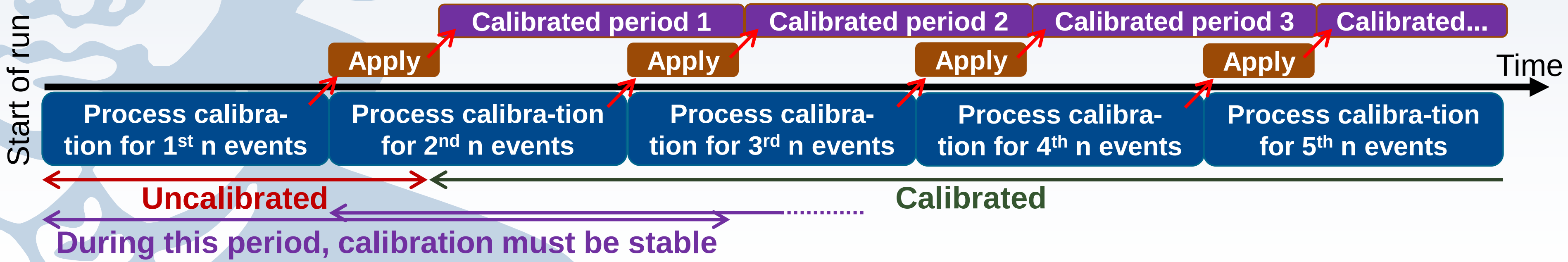
- **On the other hand, this can prohibit e.g. matching of TPC and ITS tracks, when only the TPC tracks are displaced.**



TPC z-position is obtained from the drift time.
If the drift time calibration is wrong,
the z-position of clusters is displaced.

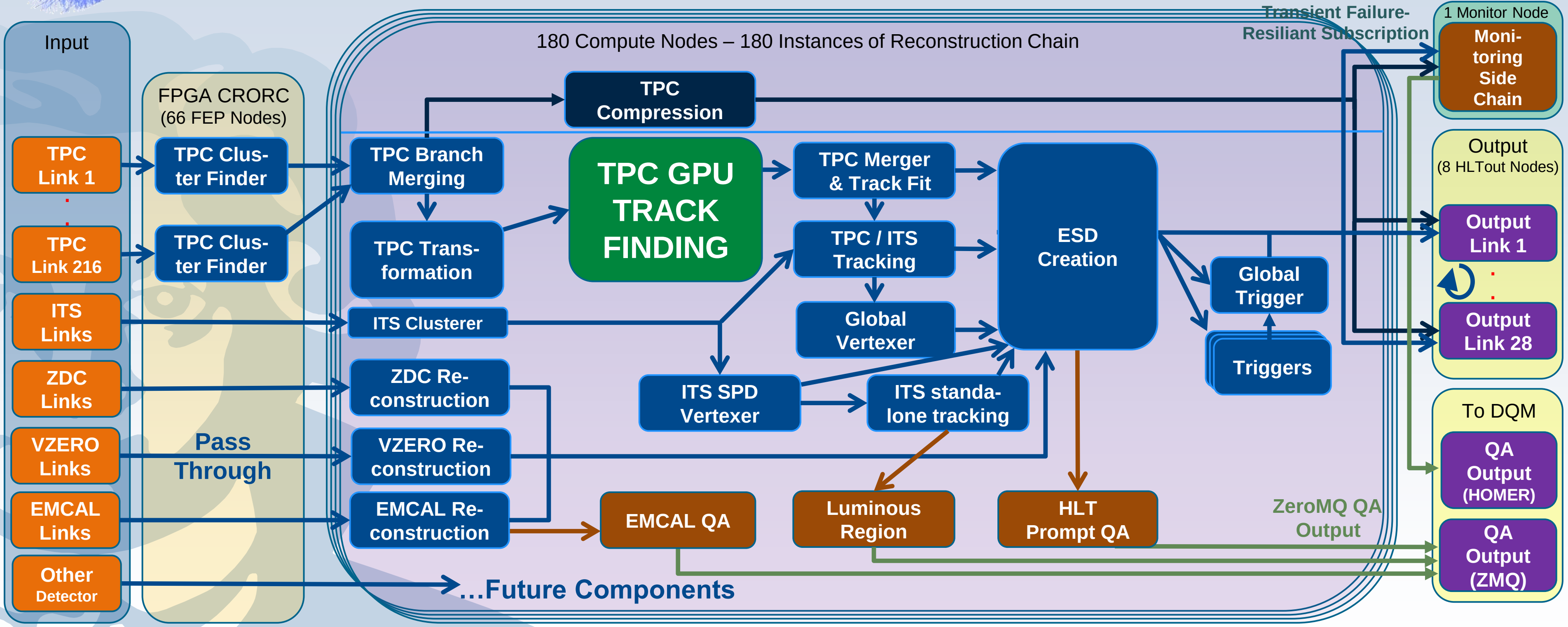
# General approach for eventual calibration

- Calibration is time-dependent – drift time is calibrated in time intervals.
- Calibration needs to process in the order of 5000 events (Pb-Pb) per time interval.
  - The calibration task is slow, calibrating 5000 events can take minutes.
  - We should reconstruct after the calibration, but we cannot cache all events that long.
    - Offline is running multiple cycles of reconstruction and calibration.
    - → There is no way we can use the calibration for the reconstruction of the first events we record.
  - But, the calibration is stable over a certain time period (here we assume 15 minutes for the drift time):
    - We can run the calibration for some time, and use the result for the next time period, as long as it is stable.
    - → **Reconstruction of the first events is not calibrated, but for all following events it is.**


Start of run
Calibrated period 1
Calibrated period 2
Calibrated period 3
Calibrated...
Apply
Apply
Apply
Apply
Time
Process calibra-tion for 1st n events
Process calibra-tion for 2nd n events
Process calibra-tion for 3rd n events
Process calibra-tion for 4th n events
Process calibra-tion for 5th n events
Uncalibrated
Calibrated
During this period, calibration must be stable

- **The important question is: Can we process the calibration for enough events and apply the result in time?**

# Overview of current HLT components



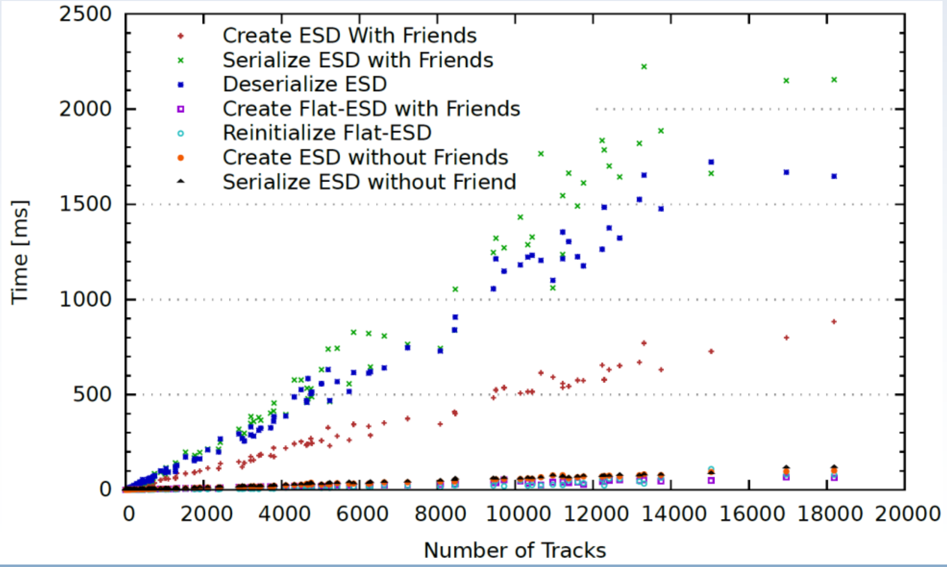
- The HLT data flow in the HLT is a directed loop-free graph.
- Events are aggregated and processed by the compute nodes in a round-robin fashion.

# Running Calibration Tasks in the HLT

- **New flat event data structure developed for HLT that enables fast data transport (Flat-ESD).**
  - Accessible similarly as the normal ESD via a virtual interface.

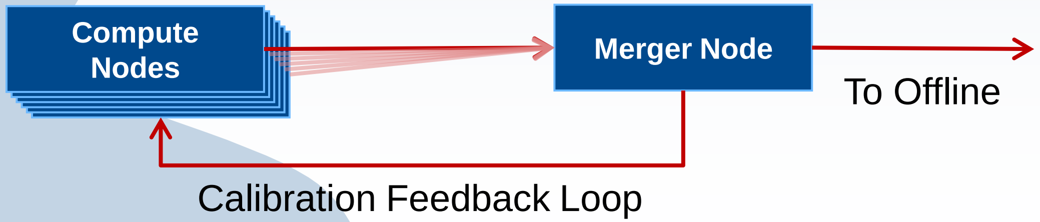
- **New HLT component can process normal ALICE analysis / calibration tasks via a wrapper.**
  - Same code online and offline.
  - Supports all analysis tasks with minor changes to support Flat-ESD (needed for fast data transport).
  - We use **TPC Drift Time Calibration** as a **first example** – scheme is extendable to other tasks.

- **Creation / Serialization of Flat-ESD much faster than for normal ESDs.**
  - (Friends are an additional structure with information needed for calibration).
  - Usage of normal ESDs with friends infeasible in the HLT.



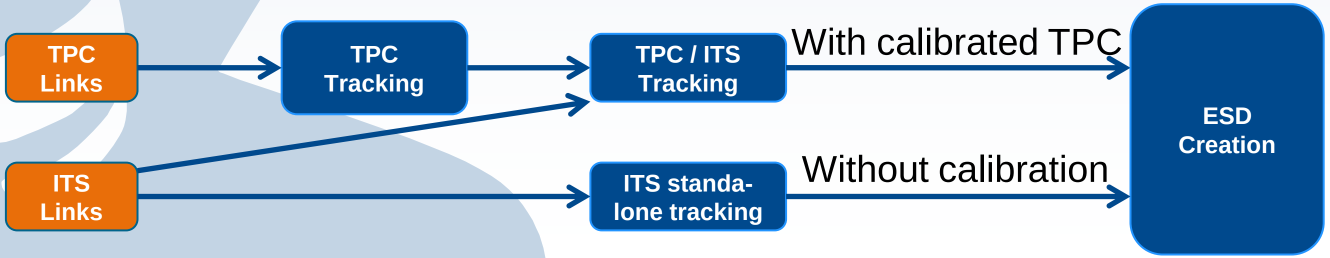
# Data Transport via ZeroMQ

- **New data transport for online calibration uses Zero-MQ**
  - We cannot use the normal HLT framework for the feedback loop, because the normal HLT data transport must be loop free.
  - We want to use an asynchronous transport, such that a failure does not affect data taking.
  - **ZeroMQ** is one of the supported backends of **FairMQ**: The data transport framework designated for $O^2$ (ALICE Run3 Upgrade)
    - It uses a multi-part message queuing transport approach like in the HLT and as foreseen for $O^2$.
    - This multi-part message queue approach for new HLT developments is a prototype for the data layout and messaging in $O^2$.
    - Zero-MQ is also used as a new transfer method to provide histograms to DQM.
- **Calibration needs more events than a single compute nodes has access to**
  - One component running per compute node, that creates objects containing all events processed on that compute node.
  - Objects are merged periodically in the background on a designated compute node.
  - Final objects are stored for offline used, and fed back into reconstruction.



# Approach for fast ITS Tracking

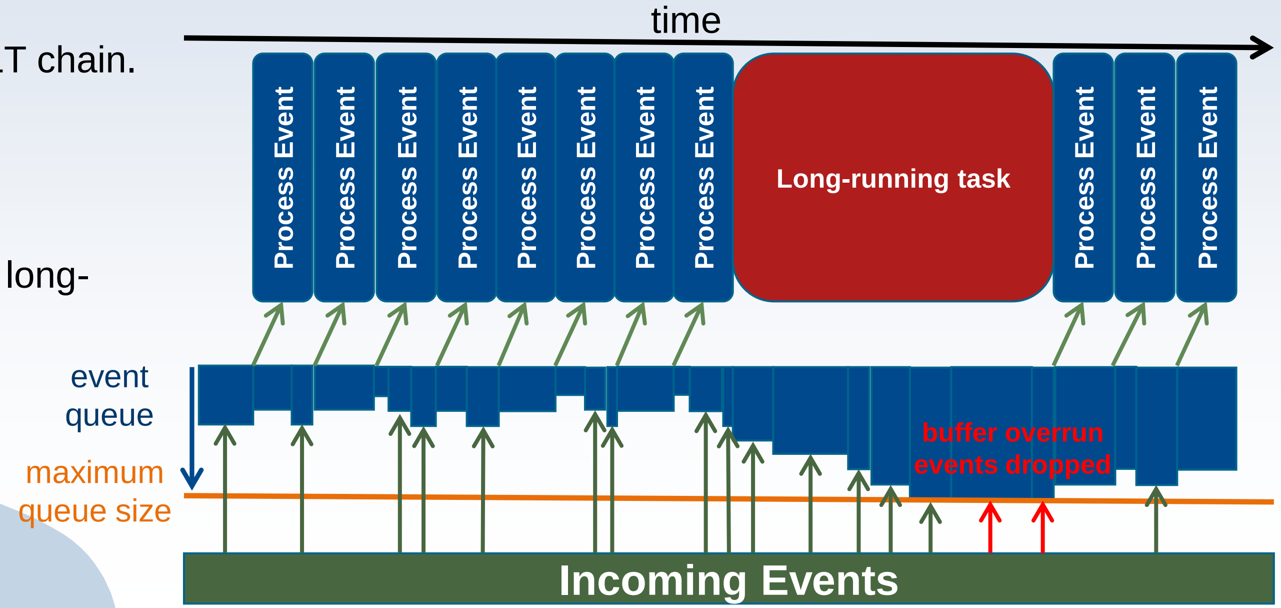
- Calibration requires TPC and ITS tracks, Luminous region needs ITS tracking (TPC is too far away and not precise enough):
  - HLT creates ITS tracks through TPC prolongation.
    (Full ITS standalone tracking is slow because of the complicated combinatorics.)
  - This might not work properly, if the TPC is not calibrated → Chicken and egg problem.
  - → We run a new fast standalone ITS tracker based on SPD tracklets.
    - **This approach reduces the combinatorics significantly, hence it is fast.**
    - **It might not find all tracks.**
    - **Still, it is sufficient for the calibration (and some other tasks).**
  - – HLT still uses TPC-ITS prolongation for full ITS tracking.



# Challenges for Online Calibration

- **HLT processing based on events.**
- **Components process one event after another.**
  - → Long running infrequent tasks could make the event buffer overrun, even if the average processing time is short.
  - → This will stall the entire HLT chain.
  - → Events will be lost.

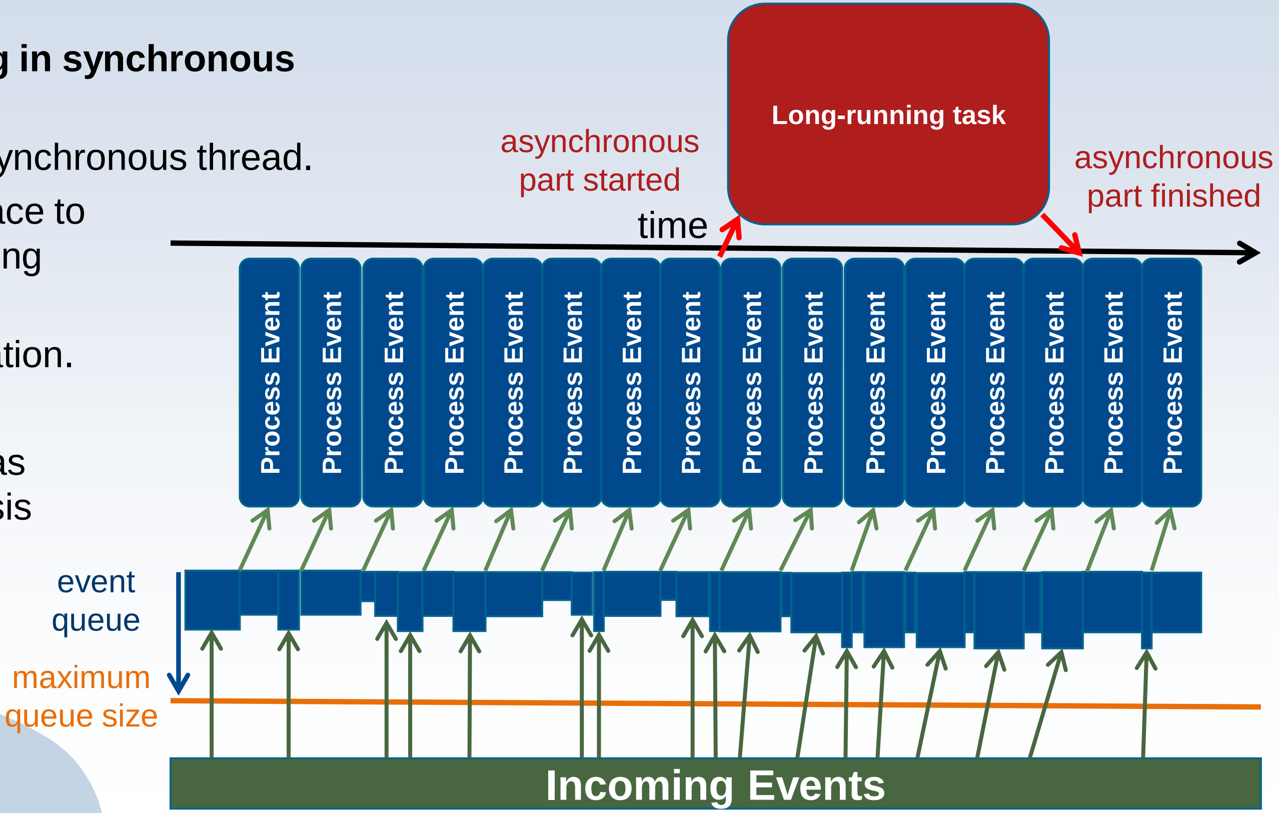
- **Examples in calibration:**
  - Accumulating events first, long-running fit later.
  - Fast event selection, long event processing.



# Asynchronous Side Tasks

- **Solution: Split processing in synchronous and asynchronous part.**
  - Framework spawns an asynchronous thread.
  - It provides a simple interface to the component for offloading asynchronous tasks.
  - It handles the synchronization.

  - Process as many events as possible on best effort basis
  - Skip remaining events.



# Asynchronous Threads / Processes

- **Processing either in asynchronous thread or asynchronous process.**
  - We do not have the calibration code under our control.
  - A segmentation fault must not stop data-taking.

- **Asynchronous thread:**
  - Fast communication via pointers.
  - Less memory consumption.

- **Asynchronous process.**
  - Communicates via shared memory, still slightly slower.
  - Segmentation fault or memory leak in asynchronous process does not stop data taking.
    → **Resiliency for fatal errors in HLT.**
    – A failing calibration component stops the calibration, but normal HLT operation continues.
  - Possibility to restart failing process..
    – (some events will be lost for calibration, processing continues transparently).

# Challenges and Solutions for Online Calibration

- Calibration involves long-running tasks, which cannot run in the HLT in an event-synchronous way.
- HLT must be resilient to failures in the calibration.

→ **Asynchronous tasks**

- HLT is loop-free, calibration is created at the end of the chain, and must be used at the beginning.

→ **Zero-MQ side-channel, that feeds back calibration asynchronously.**

- Calibration requires TPC and ITS tracks.
  - We need fast online tracking algorithms.

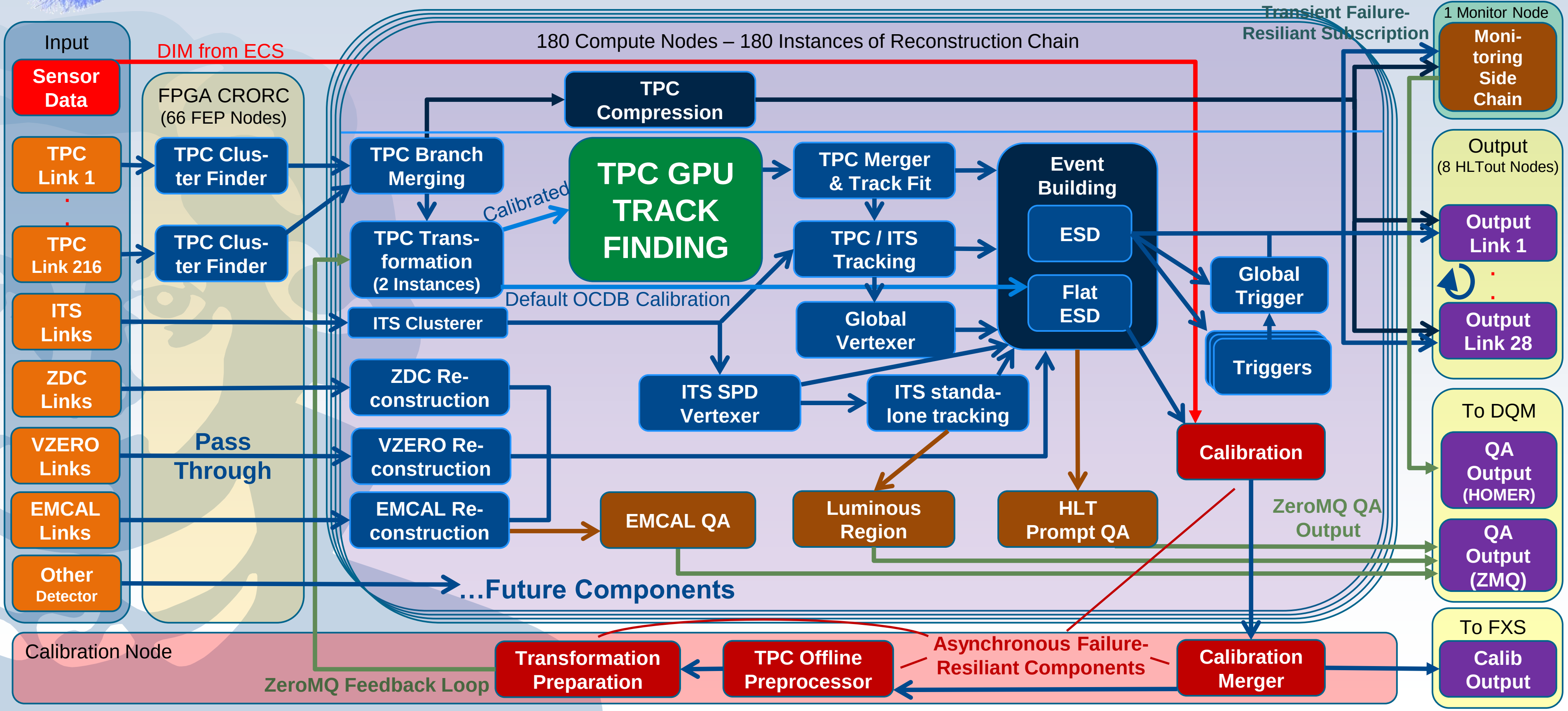
    → **GPU TPC Tracking**
  - There is a chicken and egg problem: calibration needs reconstruction and vice versa.

    → **Fast standalone (simple) ITS tracking to prepare calibration.**

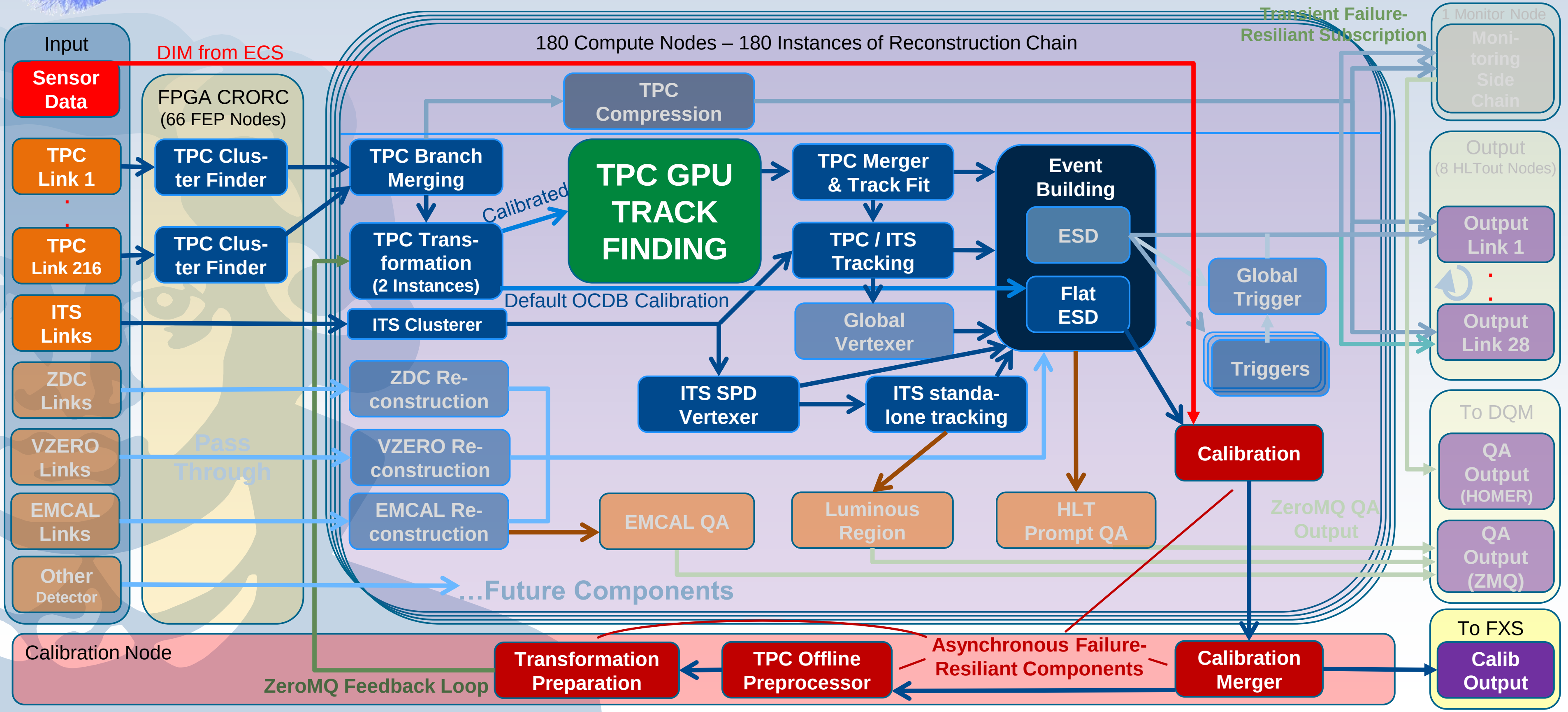
- Calibration must process in the order of 5000 events (in Pb-Pb), which takes some time.
  - We cannot cache events that long.

    → **Apply the calibration after some delay as long as it is stable.**
    (The first events of a fill are processed online without calibration.
    For offline, the full calibration is available.)

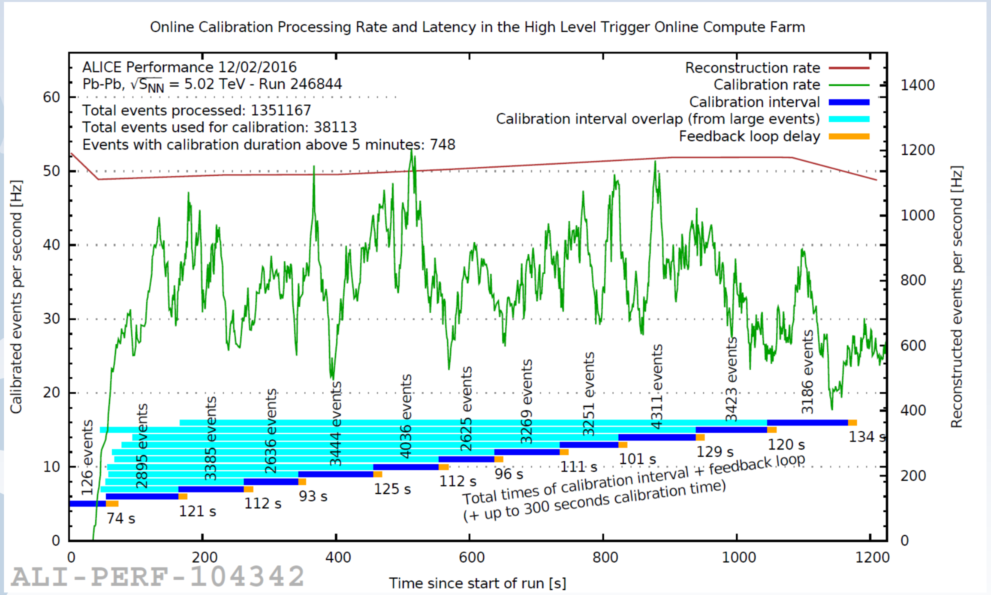
# Overview of current HLT components



# Overview of HLT calibration components.

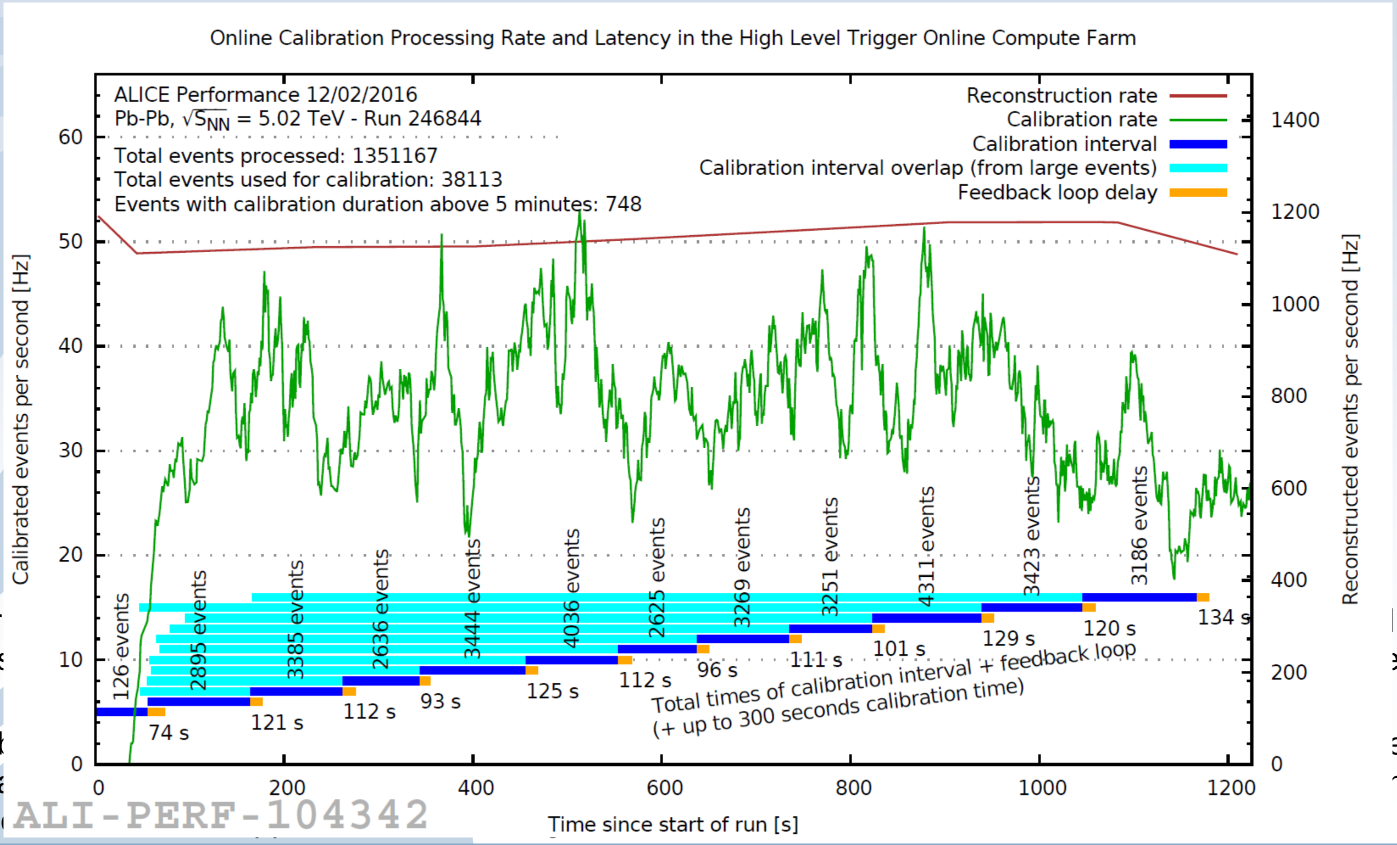


# Online Calibration Results (TPC Drift)



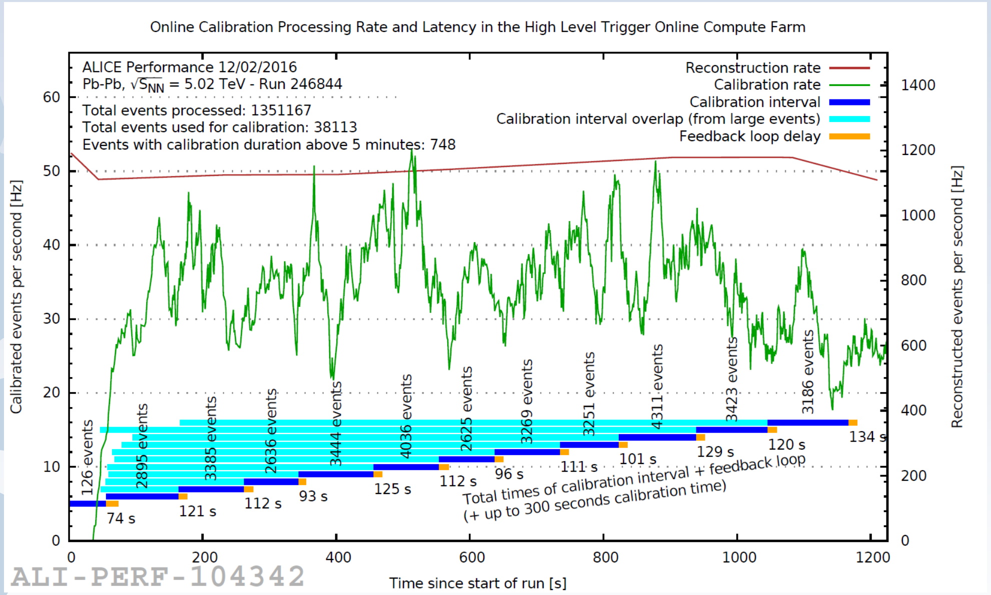
- The calibration rate shows the total number of events that finished calibration per second aggregated over all HLT nodes. The rate is 0 at the beginning because at t = 0 the calibration for the first events is already running but no events are finished yet.
- The dark blue bar shows the duration of one calibration interval and the number above states the number of events that finished calibration during that calibration interval. The yellow bar is the additional delay after the calibration interval ends up until the new calibration is applied in HLT tracking.

# Online Calibration Results (TPC Drift)



- The calibr… HLT nodes. The rate is … nts are finished yet.
- The dark b… events that finished ca… nterval ends up until the n…

## Online Calibration Results (TPC Drift)

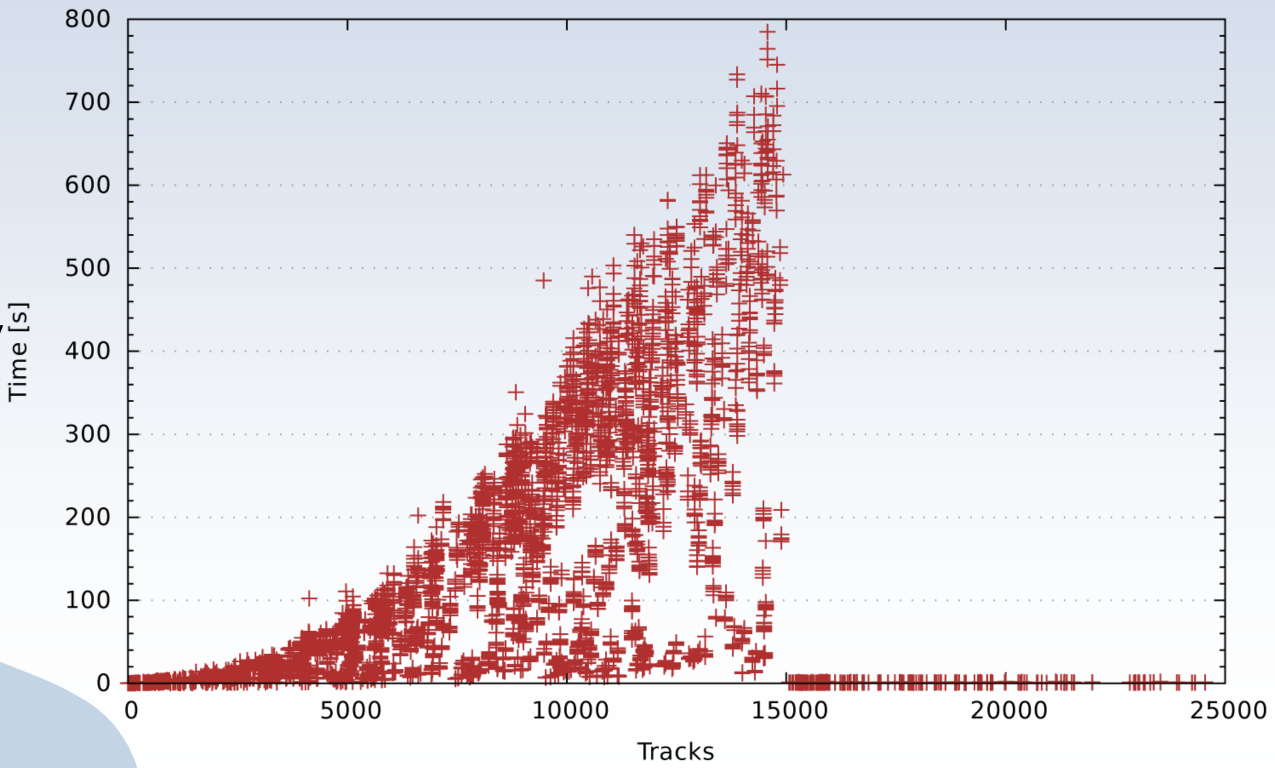


- Due to the long-running calibration task, the calibration of an event that finishes during one interval can start before the beginning of that interval. The light blue bar shows when the first event was recorded that ends in the calibration interval. The processing of an event can take up to 15 minutes, but only around 2% of the events take more than 5 minutes.
- Overall, the plot shows that the HLT can run the calibration tasks on events at an average of 31 Hz and it can apply a calibration that contains 98% of the processed events to the reconstruction with a delay of less than 7 minutes. (up to 5 minutes calibration time + ca. 2 minutes calibration interval and feedback loop delay)

# Calibration Processing Time

- Calibration processing times goes roughly quadratically with the number of tracks. (combinatorics of track matching goes quadratically.)
- The calibration applied several cuts, e.g. for less than 15000 tracks.
- More cuts are applied during the processing, leading to many events that finish early.
- Maximum processing time is 800 seconds.

- It makes sense to reduce the track threshold for online calibration.
- We are also working on speeding up the procedure in general.



# Online Calibration Results

- **First successful test of online TPC drift time calibration during Pb-Pb data taking under real conditions in December 2015.**
  - 5 CPU cores per node (on 120 nodes) used for calibration → Processing rate: 31 Hz.
  - New optimized configuration achieves 81 events per second.

- **We have seen that in principle we can run calibration tasks in a high load situation during Pb-Pb data taking online and apply the calibration results fast enough.**
  - Calibration objects passed to cluster transformation in calibration interval via feedback loop:
    - Delay: ~140 seconds (20 seconds for feedback loop and preparation of transformation map).
    - With precalculated transformation map, the calibration can be switched instantly.
  - ~ 3000 events per interval
    - 98% of events finish calibration within 5 minutes

  → Total Delay until application of transformation map: < 7.3 minutes.

- **Further online calibration tests have been running smoothly during pp data taking in 2016.**
  - Compute load in pp much less: < 3 cores per node.
  - Investigation of results ongoing, e.g.:
    - Differences in HLT and offline tracking lead to different calibration results.
      - Possibly depending on TPC occupancy.
    - Some post-data-taking configuration objects are not available during HLT processing.
    - Online calibration allows us to check online reconstruction results with higher resolution.
      - This already unveiled some bugs, which were before attributed to missing calibration.