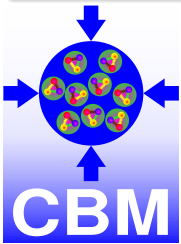
Online-Offline Data Processing Software Integration

46th CBM Collaboration Meeting



Sergei Zharko

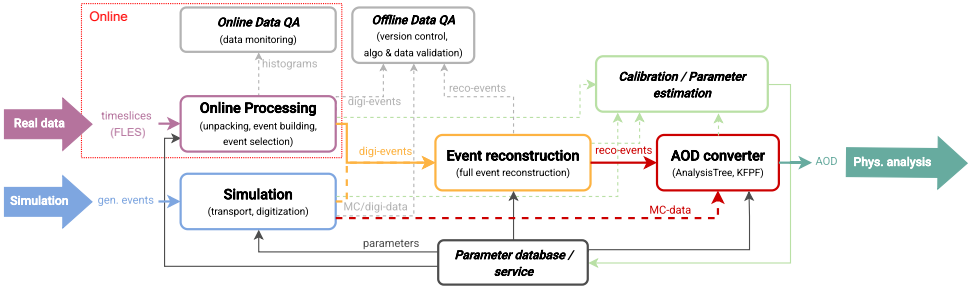
CBM Department
GSI Helmholtzzentrum für Schwerionenforschung
Darmstadt, Germany

October 22nd, 2025

Outline

- Data processing framework
- Online data processing
- Online data processing scenario examples
- Common online/offline reconstruction software
- Physics event selector
- Current status and plans

Data Processing Framework



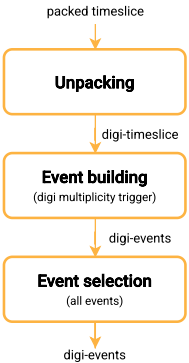
- **High Performance Computing (HPC)**:
  - Timeslice/event reconstruction algorithms
  - Online processing (OP) software
  - Performance-critical: strict requirements to algorithms, data-classes and external tools
- **Offline**:
  - MC-simulations
  - Detector calibration and alignment
  - Parameter estimation
  - Physics analysis
  - Precision and flexibility: softer requirements

Online Data Processing Software

- **Goal:** reduction of data rates with software triggers and selectors for interesting/rare events
- **Tasks:**
  1. **Build** events using **time/phase-space triggers** in continuous data stream
  2. **Select** events, which satisfy a specific selection criteria
- **Data-processing steps:**
  - **Unpacking**: conversion of FEB messages to digitized detector signal (**digis**)
  - **Timeslice local reconstruction** in different detector modules (**hits** generation)
  - **Timeslice global reconstruction**: tracking and primary vertex finding
  - **Event building**: grouping digis into **digi-events** based on the configured trigger
    - **Different triggers can be used**: digi multiplicity trigger, hit-multiplicity trigger, PV-trigger, ...
  - **Event reconstruction**: local + global reconstruction within a digi-event
  - **Event selection**: selection of a digi-event for storing
    - **Different selectors can be used**: digi/hit/track multiplicity, rare particle signals, decay topologies, ...
- **OP Software:**
  - **cbmreco** – a monolith[1] configurable process, multiple processes run on different nodes
  - HPC algorithms:
    - new unpackers and hit-finders, CA-tracking
    - CPU-parallel
    - GPU-parallel: STS hit-finding, CA (talk by Grigory Kozlov)
    - no dependency on simulation software and (almost) on ROOT
    - KFParticleFinder: available for implementing physics selectors
  - **OP scenario (selection of data processing steps in different detectors) is defined by a trigger and selector**

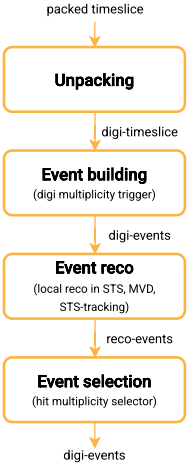
[1] Currently, alternative implementation variants are under discussion (talk by Bartosz Sobol)

Online Processing Scenario Example: Min. Bias



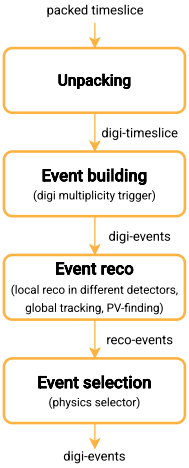
- Minimal scenario for OP (no data-reconstruction)
- Stores all digi-data (digi-events, built with a digi mult. trigger) without selection
- Suitable for interaction rates at ∼100 kHz (Day-1)
- Subject of the upcoming VT25 (talk by Volker Friese)

Online Processing Scenario Example: Digi-multiplicity Trigger & Hit-multiplicity Selector



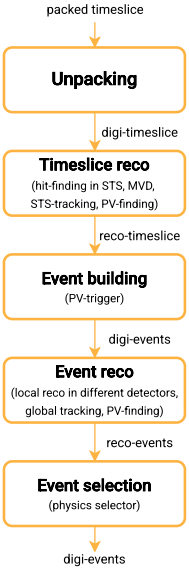
- Adds a simple selector, e.g. hit multiplicity in STS and TOF
- Involves local event-reconstruction in the corresponding subsystems

Online Processing Scenario Example: Digi-multiplicity Trigger & Physics selector



- Requires full event-reconstruction (global tracking, PV-finding)
- **Requires alignment of geometry**
- **Usage of detector subsystems in reconstruction** is defined by particular event-selector
- A subject of the Day-2 and the vertical tests in the beginning of 2026

Online Processing Scenario Example: PV-trigger & Physics Selector



- Involves partial timeslice-reconstruction: tracking in STS+MVD, PV-finding

Common Online/Offline Reconstruction Software: Status

- Reconstruction algorithms – **common** data-processing step in online and offline:
  - requires **common algorithms**
  - requires **common parameter- and data-formats**

- At the moment – two code bases of reconstruction algorithms in CbmRoot:
  A. **Offline**:
     - hit-finders (FSD, MUCH, MVD, RICH, STS, TOF, TRD, TRD-2D) (**legacy**)
     - STS-tracking (CA)
     - global-tracking (LIT) (**legacy**)
     - PV-finder (**legacy**)
     - Old data-format (inherit ROOT `TObject`, store data in `TClonesArrays`, incompatible with GPU)
  B. **Online**:
     - hit-finders (BMON(**simplified mCBM version**), STS, TOF, TRD, TRD2D)
     - STS-tracking (CA)

- **Only one common online/offline code base should stay**:
  - **porting to online is straight forward**
  - required: proper configuration and parameter definition
  - required: new data-format for reconstructed data to meet different use-cases
  - global tracking: implementation within CA (talk by Sergey Gorbunov)

Common Online/Offline Reconstruction Software: Current Developments

- New format for reconstructed data (**WIP**)
  - classes for hits in different detector subsystems, track, vertex, PID-input and matching
  - non-polymorphic (no virtual functions, ROOT, inheritance of TObject, etc.) – GPU friendly
  - self-contained `RecoEvent` and `RecoTimeslice`
  - **common for online/offline**

- **RecoSetup** – a representation of CBM-setup in reconstruction algorithms (**WIP**)
  - properties of subsystem modules needed for local reconstruction (physical, geometry, addressing)
  - properties of different subsystems as tracking detectors (material budget, geometry, addressing) – replacement for tracking detector interfaces
  - **modular, has no dependencies on the origin of parameters**

- **Event reconstruction module** – a full event reconstruction class (**WIP**):
  - uses the same algorithm, data- and parameter-formats as in cbmreco
  - solid: all steps of the event reconstruction are included, selection of the steps is defined by configuration and geometry
  - uses well-defined input (**DigiEvent**) and output (**RecoEvent**)
  - prospects for parallelization on CPU and GPU

- **Parameter service** – reconstruction parameters generation on request (**WIP**)
  - has access to the full parameters database
  - generates a consistent and valid set of parameters (geometry, calibrations, module properties) for online data processing
  - beta version is available (the `cbm-online-par-dump` binary)
  - to be cleaned from hard-coded parameters for mCBM

Physics Event Selector

- A **generic event-selector**:
  - functor:
    $$\text{Select?} = f_{\text{Ev.Sel.}}(g_{\text{RE-Converter}}(\text{RecoEvent}), \text{parameters}, \text{configuration})$$
  - input: **a converted RecoEvent** (properties of trajectories, associated PID-input measurements)
  - input: **a set of pre-estimated parameters** (fit, cuts, ML model settings)
  - input: **configuration** – list of active detector subsystems
  - output: **a boolean** – selection result
  - **To be prepared and tested by a corresponding PWG** (a QA-module is to be provided as well)

- Requirements for event-selectors from the framework:
  - **API**: format of the pre-estimated parameters structure
  - **Performance**: the implementation must satisfy defined limits on execution time and resource usage

- **KFParticleFinder-based** event-selector (as a subset of generic event-selectors):
  - reliable decay topology reconstruction package, tested in CBM, ALICE, STAR (talks by Iouri Vassiliev and Yingjie Zhou)
  - **the KFParticleFinder is already available in cbmreco**
  - **requires a uniform way of providing PDG-info to define input for KFParticleFinder**:
    - functor:
      $$\text{PDG} = f_{\text{PID}}(\text{track} + \text{PID-input}, \text{parameters}, \text{configuration})$$
    - input: track (trajectory properties), associated PID-input measurements
    - input: pre-estimated parameters from **selected** PID frameworks
    - input: configuration – list of active detector subsystems
    - output: PDG-code, associated with input track

- **Framework implementation details for physics event-selector are under ongoing discussion**

Summary: Current Status and Plans

- **Missing reconstruction algorithms (common online/offline)**:
  - local reconstruction: BMON, MVD(**WIP**), RICH(**WIP**, talk by Martin Bayer), MUCH, FSD, MUST(**WIP**)
  - global reconstruction: global tracking and primary vertex finding
  - **most of these algorithms can be ported from legacy ones by applying new format of reconstructed data and proper parameter handling**

- **WIP**:
  - new **online/offline** format for reconstructed data (the RecoEvent class, classes for track, vertex, hits, PID-input measurements and matching info)
  - CBM setup representation in reconstruction algorithms
  - event-reconstruction module
  - parameter service developments

- **Upcoming developments**:
  - Event selection framework
  - Porting reconstruction algorithms
  - Integration of online QA classes into offline QA (as a subset)

- **Features needed**:
  - proper definition of software address (`CbmAddress` class): BMON, MUST
  - online QA for BMON, MVD, RICH, MUCH, FSD, TRD, MUST
  - offline QA for BMON, FSD, MUST

Thank you for your attention!