

Computing Challenges and Developments for CBM

Volker Frieese



Helmholtzzentrum für Schwerionenforschung

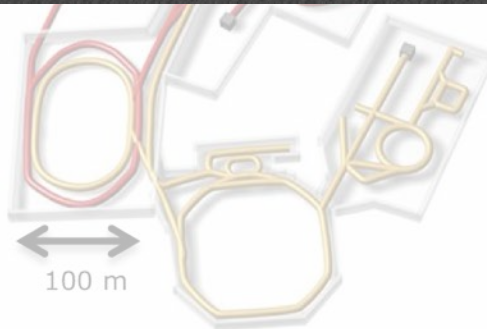
Guwahati, 28 September 2019

CBM – A Big Data-Producer



Primary Beams

- $10^9/\text{s}$ Au up to 11 GeV/u
- $10^9/\text{s}$ C, Ca, ... up to 14 GeV/u
- $10^{11}/\text{s}$ p up to 29 GeV

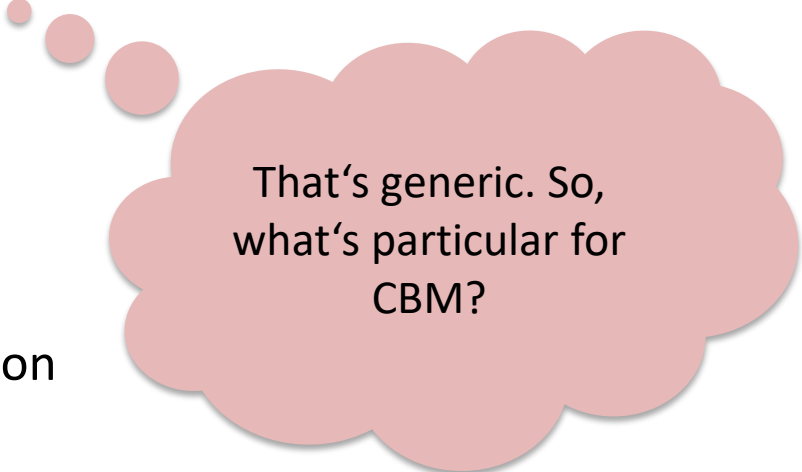


FAIR phase 1
FAIR phase 2

Computing in CBM

CBM Computing has to provide the software and related tools required to

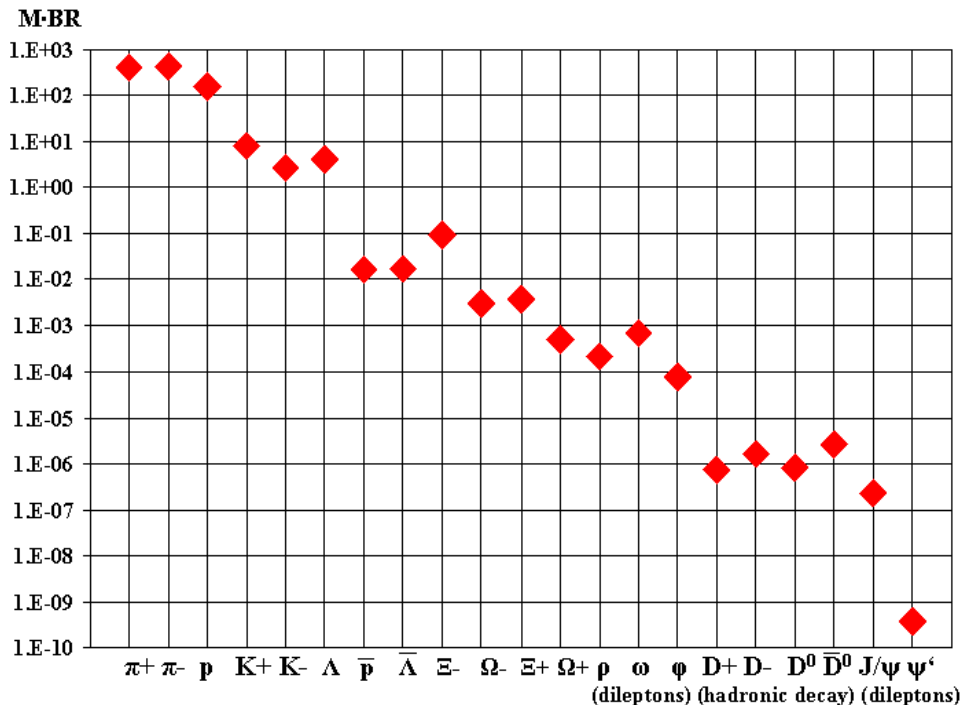
- operate the experiment
 - configuration, control, DAQ, online data processing, data storage
- analyse data
 - reconstruction, PID, data access
- simulate the detector setup
 - detectors, electronics, data acquisition



That's generic. So,
what's particular for
CBM?

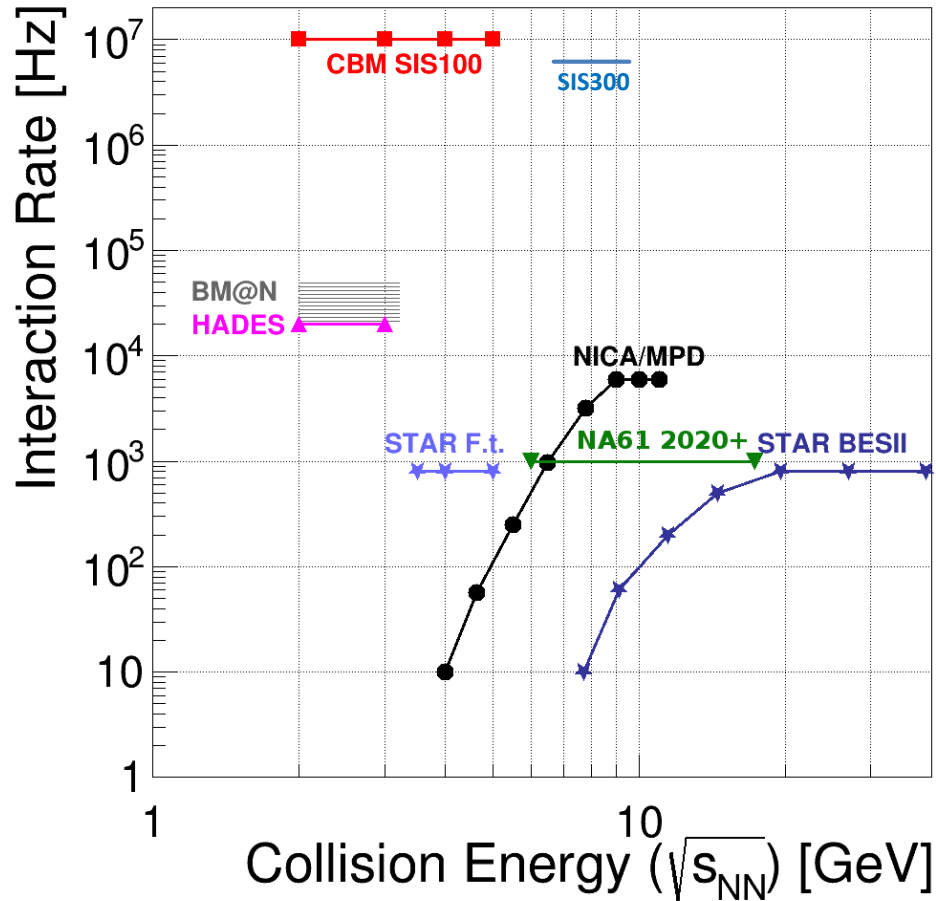
Rare Observables

Model predictions of particle multiplicities (x branching ratio)
(central Au+Au, 25A GeV)

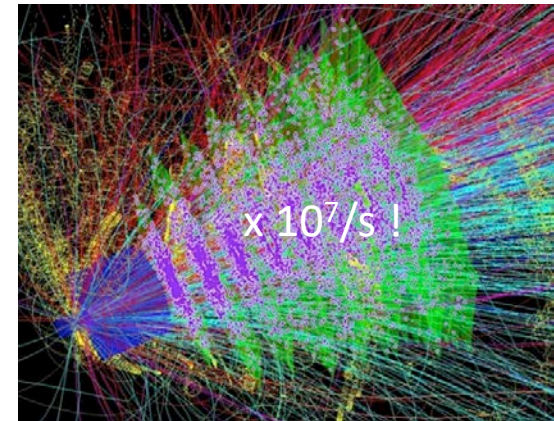


- Some of the (most interesting) probes are extremely rare.
- Decent measurement in reasonable time necessitates high interaction rates.
- Current heavy-ion experiments run with very moderate rates (100 Hz - several kHz).
- CBM targets for 10 MHz

CBM in the experimental landscape



Uniqueness of CBM: very high rate capability



Comes with huge challenges in terms of:

- Speed and radiation hardness of detectors and read-out electronics
- Data processing on- and offline

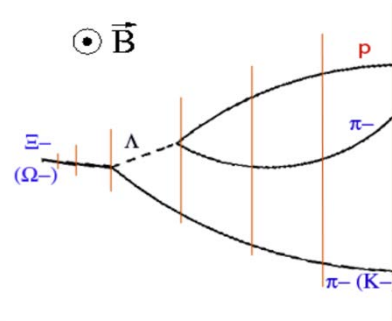
Data Rates

- Raw data event size: 100 kB / min. bias event (Au+Au)
- At 10 MHz event rate: raw data rate 1 TB/s
- Archival rate:
 - technologically possible are rates of 100 GB/s and above
 - limiting factor are the storage costs
 - typical runtime scenario 2 effective months / year (5×10^6 s)
 - At 1 GB/s: gives a storage volume of 5 PB/year

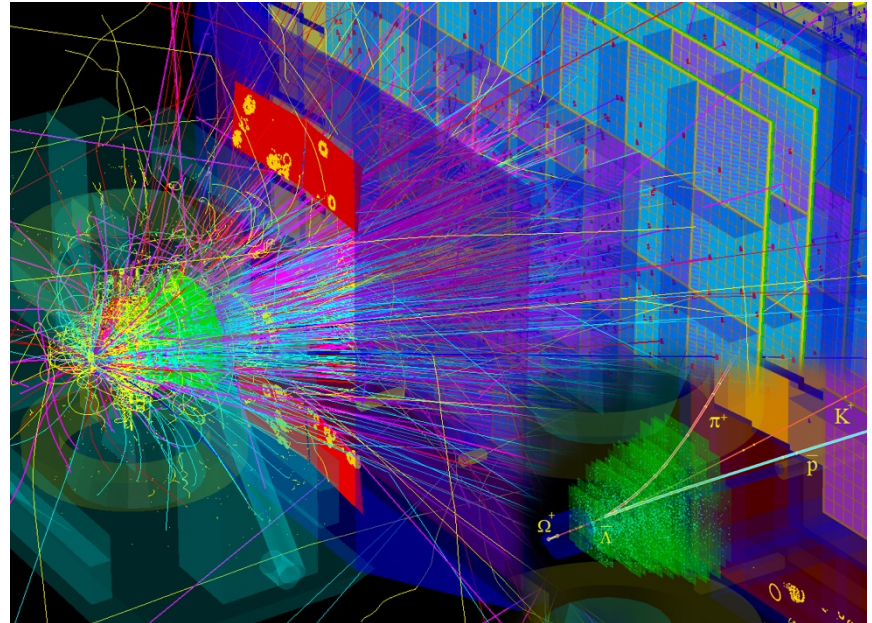


We aim at an data archival rate of a few GB/s, meaning that the raw data volume has to be suppressed online by factors 300 - 1000.

Selecting Data Online

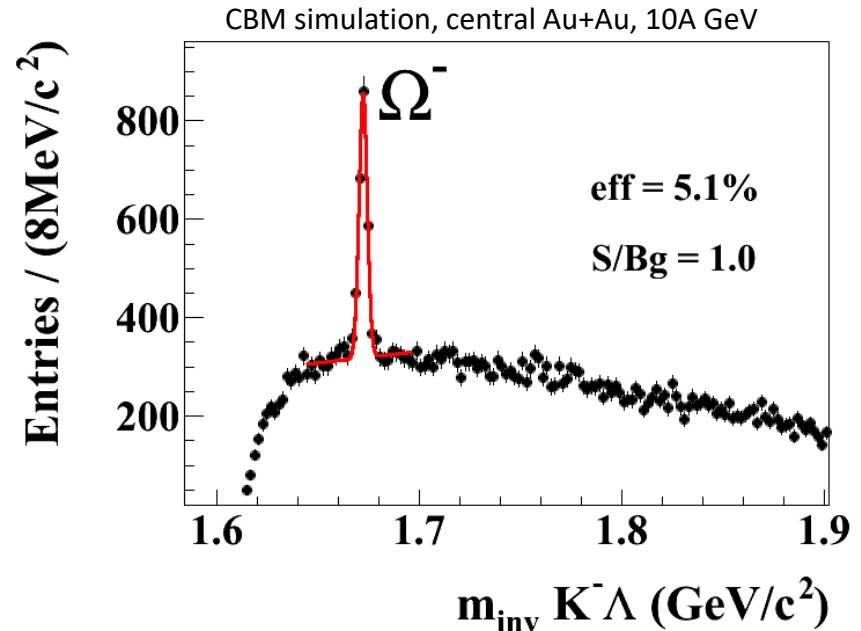


- Some (not all) of the rare probes have a complex signature.
Example: $\Omega \rightarrow \Lambda K^+ \rightarrow p \pi^- K^+$
- In the background of several hundreds of charged tracks
- No simple primitive to be implemented in trigger logic



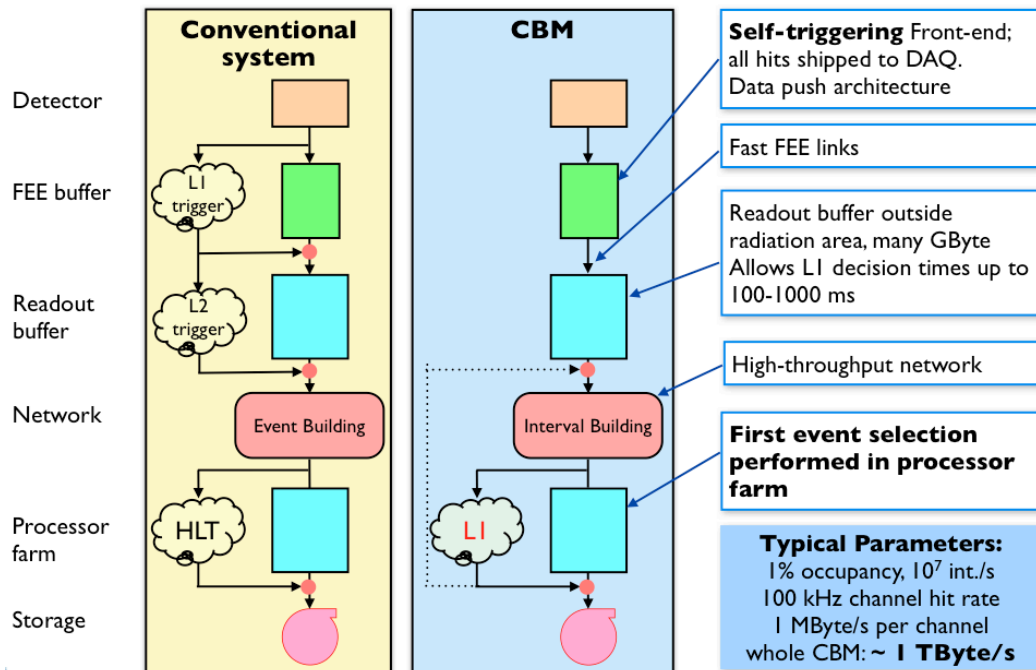
Selecting Data Online

- Selection requires reconstruction of all tracks plus combinatorial search for two decay vertices: typical software task
- Offline performance for Omega: $S/B \sim 1$
- If realisable online: excellent software trigger



- Similar argument for many topology-based observables (hyper-nuclei, exotic strange objects, charm)
- Simpler patterns e.g. for lepton pairs (J/ψ or low-mass)
- R/O design must be based on the most challenging case

DAQ and Trigger Concept



- No hardware trigger at all
- Continuous readout by autonomous FEE
- FEE sends data message on each signal above threshold (“self-triggered”)
- Hit message come with a time stamp; readout system is synchronised by a central clock
- DAQ aggregates messages based on their time stamp into “time slices”
- Time slices are delivered to the online computing farm (FLES)
- Decision on data selection is done in the FLES (in software)

Triggered and Free-Running Readout

Hardware triggers: snapshots of the detectors



A trigger-less readout: a movie of the detector



Advantages

- no latency issues; the system is limited by throughput
- no buffers on FEE ASICS (inside radiation zone) needed
- data selection is shifted to software
 - in principle, everything which is usually done in the offline analysis can be implemented for online data selection
 - very flexible: easy to switch between triggers, to use different triggers in parallel
 - assessing the trigger efficiency is straightforward: no emulation of trigger logic needed

So, why was it not done before?

- Requires an online compute farm powerful enough to process the entire data stream
- Throughput is defined by the size of the compute farm and the speed of the algorithms.
- CBM estimate: equivalent to $\sim 10^5$ CPU cores needed
- Some years ago, this was the entire LHCgrid
- Nowadays (let alone in some years), feasible to finance and to host close to the experiment

Issues of a Trigger-less System

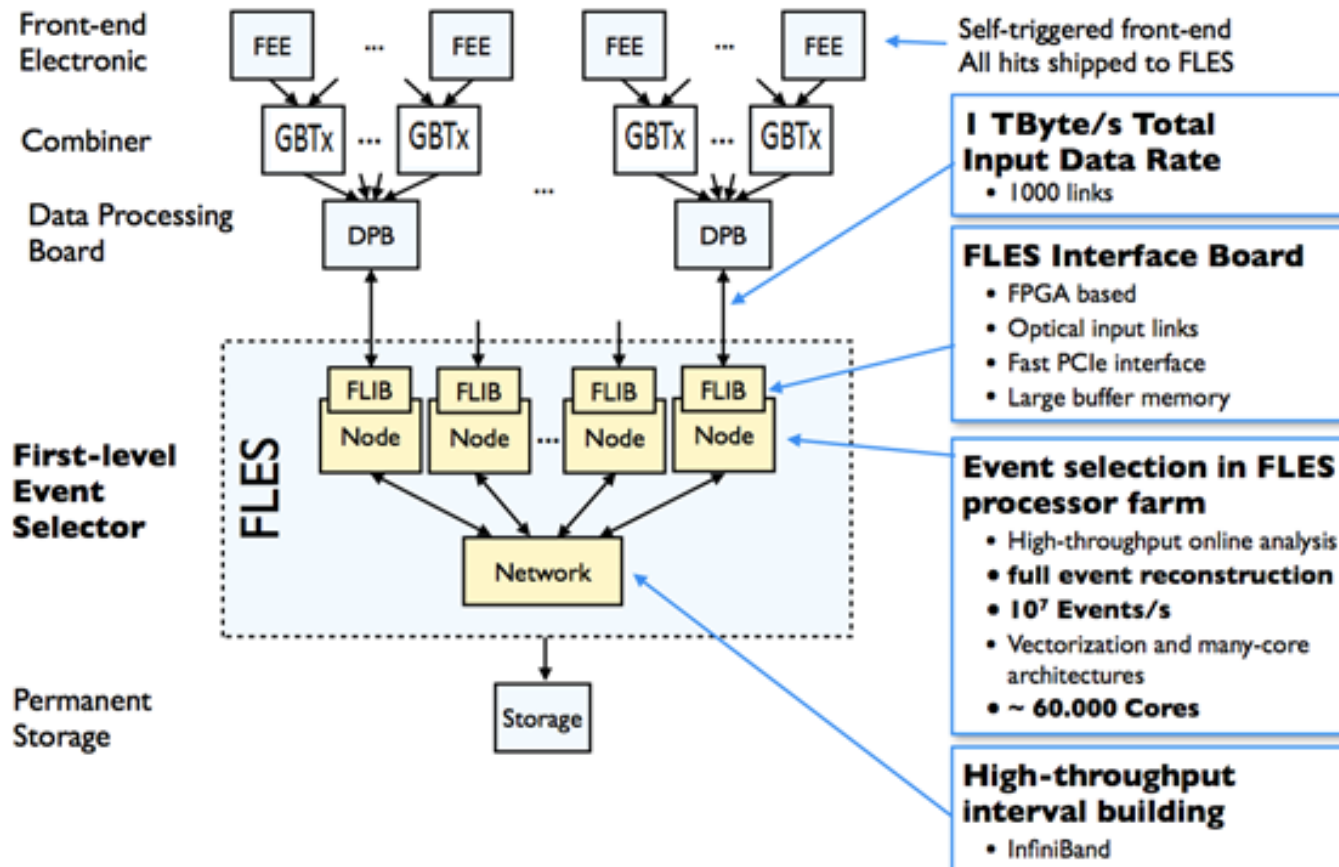
- Noise from detectors and electronics
 - tight threshold in order to suppress the contribution of noise to the total data rate
 - good signal-to-noise ratios in detectors are needed in order not to lose signals

Example: fraction of noise from the STS

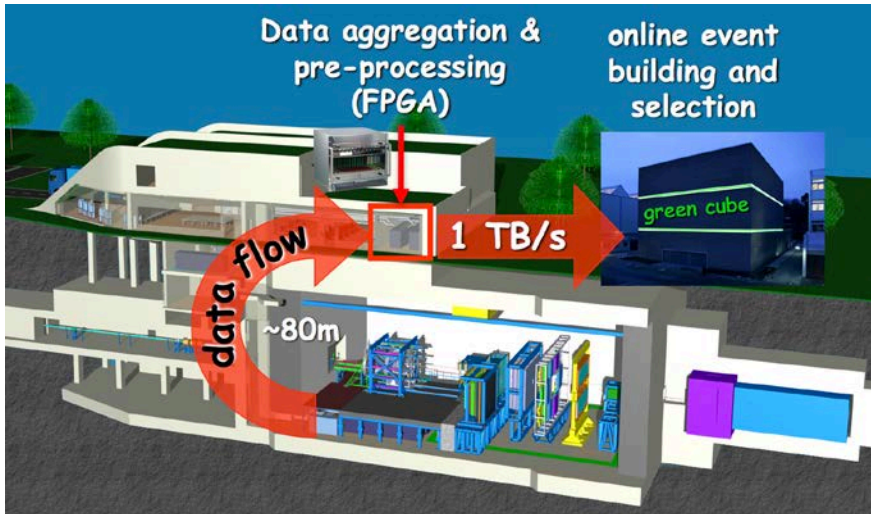
M.b. event rate	10 MHz	1 MHz	100 kHz	10 kHz
Threshold / noise = 3	40 %	86 %	98 %	99.8 %
Threshold / noise = 3.5	11 %	55 %	92 %	99.2 %
Threshold / noise = 4	2 %	15 %	65 %	95 %

- No events given to software
 - Unlike in conventional HLTs, where events are build before by DAQ
 - Online reconstruction starts from time-sorted data stream
 - Algorithms have to take into account time coordinate (“4D reconstruction”)

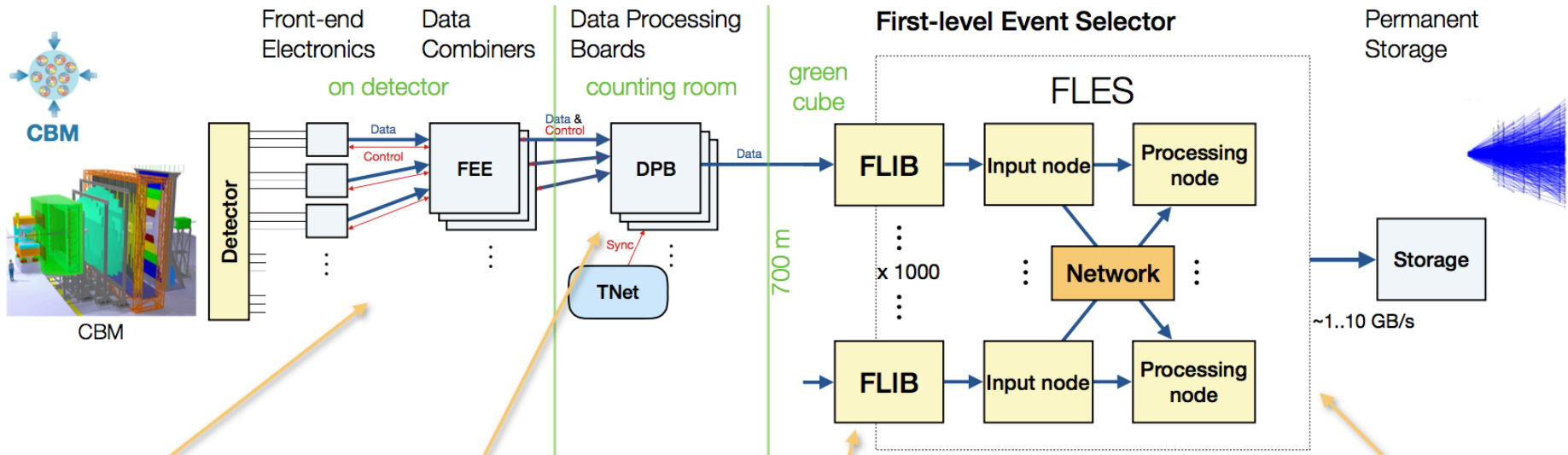
Read-out Scheme



Online Data Flow



Readout Scheme



Detector Front-ends

- Autonomous hit detection and **zero-suppression**
- Associate **time stamp** with each hit, aggregate data

Data Processing Board (DPB)

- Local data **preprocessing**: Feature extraction, time sort messages, data reformatting, merging input streams
- Convert to **global time**

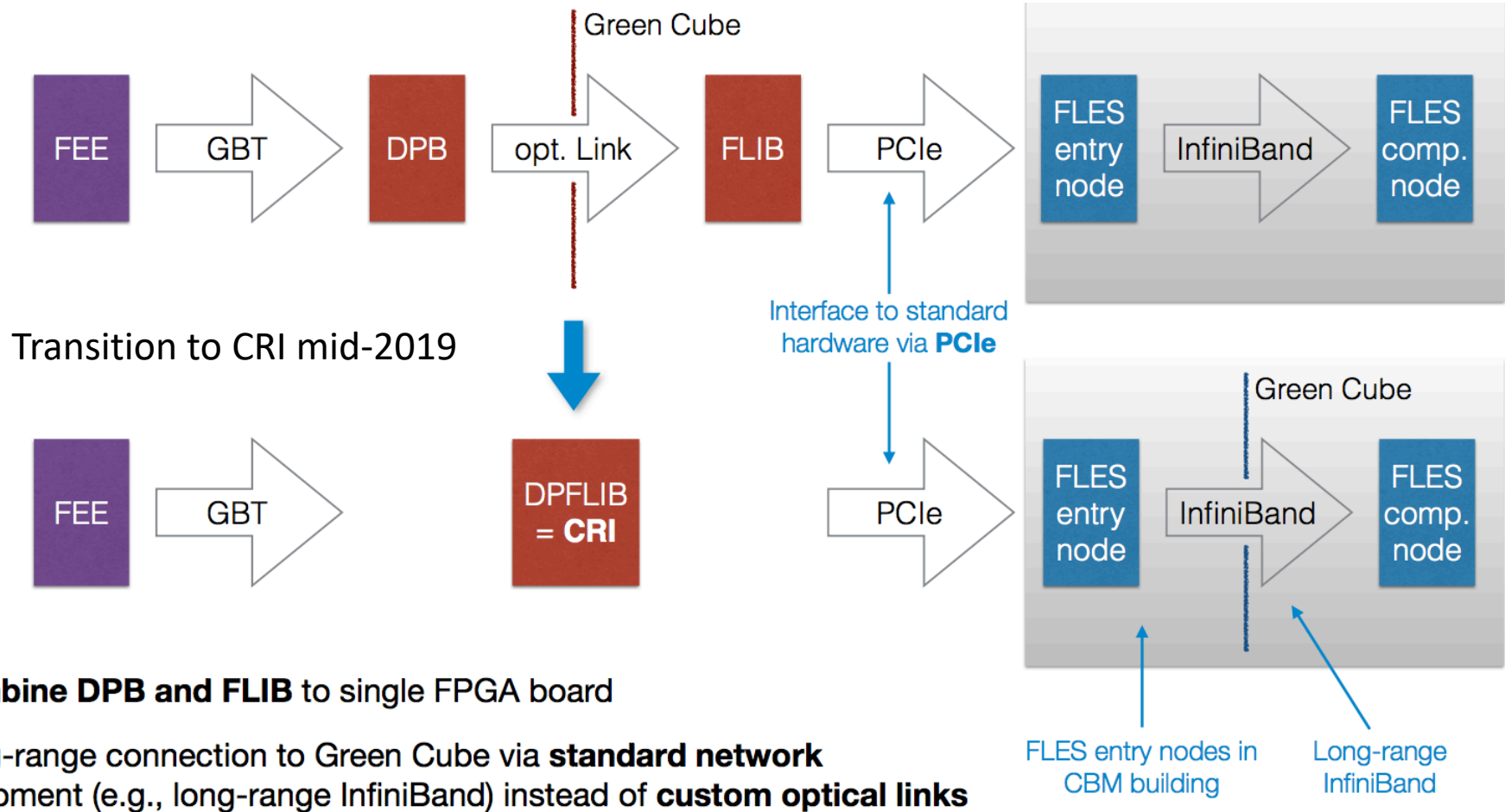
FLES Interface Board (FLIB)

- **Time indexing** and buffering of microslices

FLES Nodes

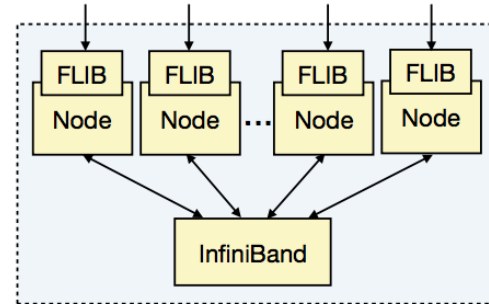
- Calibration and global feature extraction
- **Tracking in 4 dimensions** (including time)
- Full reconstruction, associate hits with events
- Identification of leptons and hadrons
- High-precision vertex reconstruction
- Event selection

Towards the Final System



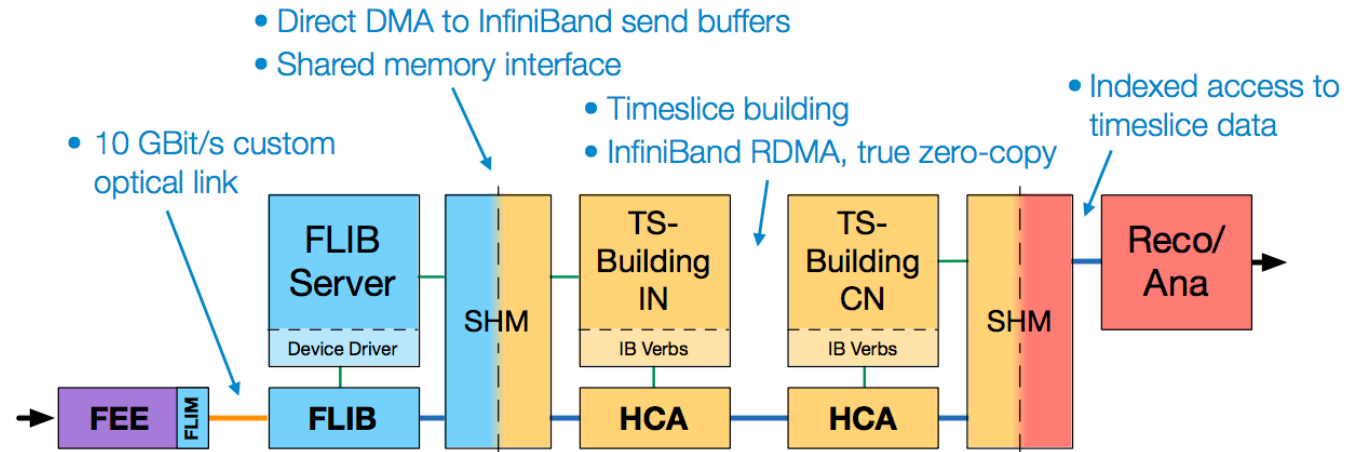
FLES Architecture

- FLES is designed as an HPC cluster
 - Commodity PC hardware
 - GPGPU accelerators
 - Custom input interface
- Total input data rate >1 TB/s
- InfiniBand network for timeslice building
 - RDMA data transfer, very convenient for timeslice building
- Flat structure w/o dedicated input nodes
Inputs are distributed over the cluster
 - Makes use of full-duplex bidirectional InfiniBand bandwidth
 - Input data is concise, no need for processing before timeslice building
- Decision on actual commodity hardware components as late as possible
 - First phase: full input connectivity, but limited processing and networking



FLES Data Management

- RDMA-based timeslice building (*flesnet*)
- Works in close conjunction with FLIB hardware design
- Paradigms:
 - Do not copy data in memory
 - Maximize throughput
- Based on microslices, configurable overlap
- Delivers fully built timeslice to reconstruction code



- Prototype implementation available
 - C++, Boost, IB verbs
- Measured flesnet timeslice building (8+8 nodes, including ring buffer synchronization, overlapping timeslices):
 - ~5 GByte/s throughput per node
- **Prototype software successfully used in several CBM beam tests**

Data Transport

Transport	Hardware	Performance
RDMA/ Verbs	InfiniBand	high
	RoCE	high
	TCP (via SoftiWARP)	reduced
	Ethernet (via Soft RoCE)	reduced
Libfabric	RDMA-capable	high
	any network	reduced
ZeroMQ	any TCP network	reduced
MPI	any network	high

← Standard

New: Libfabric transport

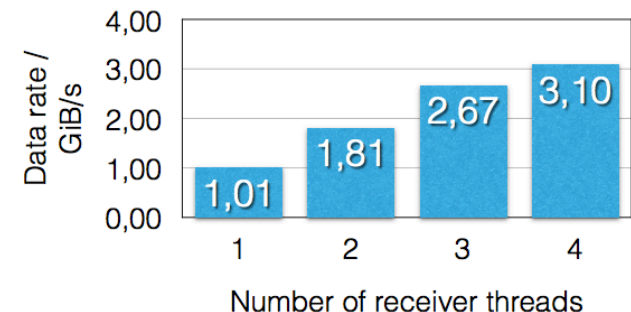
- Under active development
- Supports modern interconnect capabilities
- Libfabric micro-benchmark over 384 nodes **achieved ~646 GB/s**



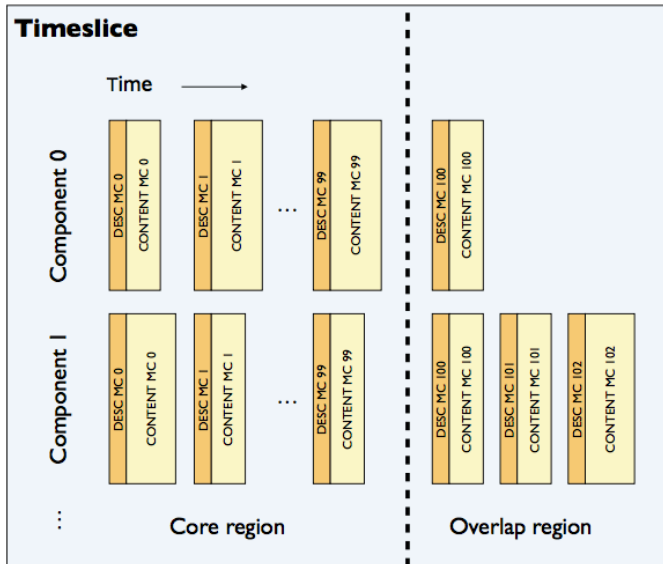
←

New: ZeroMQ transport

- Alternative for small setups



Time Slice: Interface to Online Reconstruction



Timeslice

- Two-dimensional indexed access to microslices
- Overlap according to detector time precision
- Interface to online reconstruction software

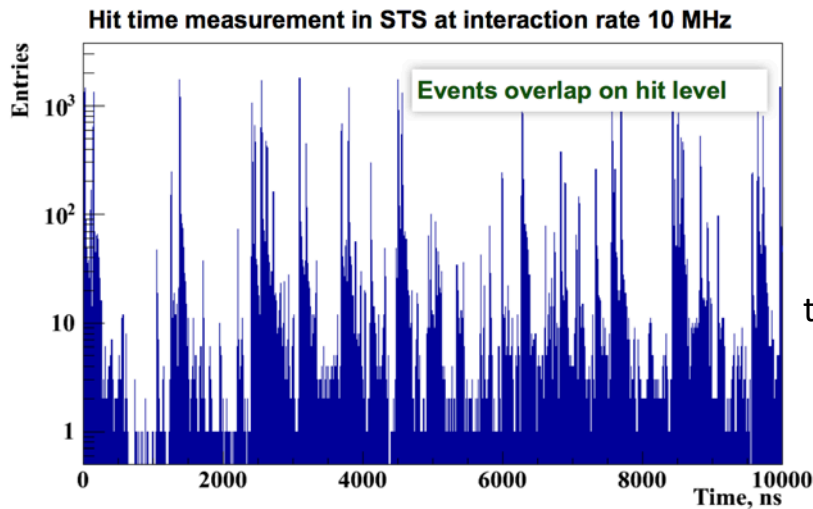
- Basic idea: For each timeslice, an instance of the reconstruction code...
 - ...is given direct **indexed access** to all corresponding data
 - ...uses **detector-specific** code to understand the **contents** of the microslices
 - ...applies **adjustments** (fine calibration) to detector time stamps if necessary
 - ...finds, **reconstructs and analyzes** the contained events
- Timeslice data management concept
 - Timeslice is self-contained
 - Calibration and configuration data distributed to all nodes
 - **No network communication** required during reconstruction and analysis

Real-Time Reconstruction

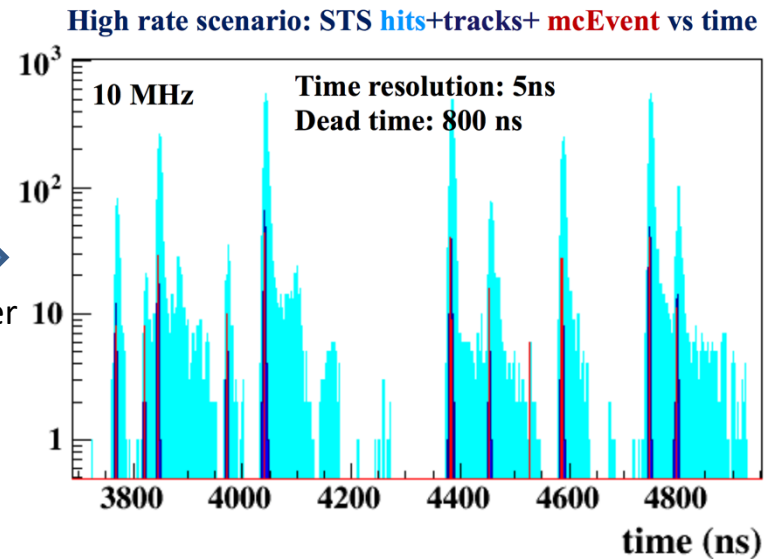
- In our concept, the task of online data selection is shifted from electronic engineering to software engineering.
- For a given event / data rate, the speed of the algorithms determines the required size of the online compute farm.
- For a given financial budget / size of the online farm, the speed of the algorithms determine the physics output of the experiment.
- High-performance online software is a pre-requisite for the successful operation of CBM.
 - Make optimal use of available parallel computer architectures: many-core, GPU, accelerators
 - Be flexible to upcoming new architectures
- Parallelism is the key word
 - Data-level parallelism: one time slice per compute node
 - Task-level and data-level parallelism within time slice

Track Finding

- Usually, the most compute-intensive task in reconstruction
- Approach: Cellular Automaton, operating on time-ordered stream of detector hits (no event association)



track finder



- After track finding, events can be defined as time-clusters of tracks

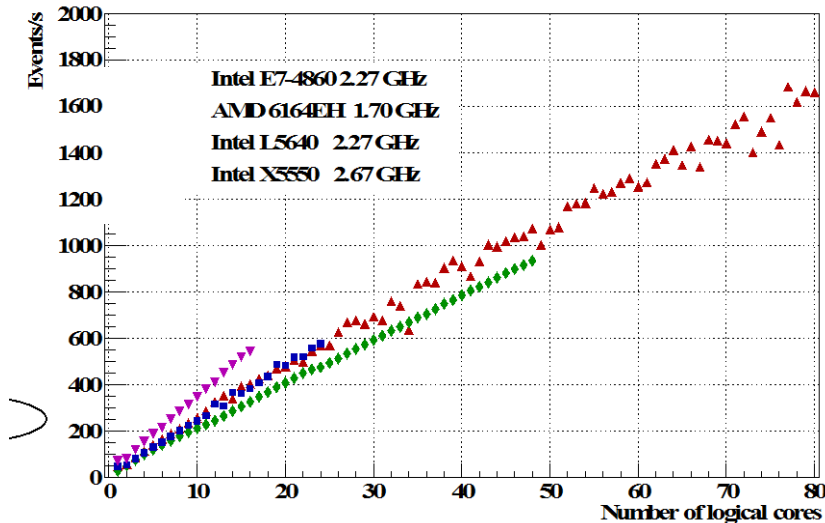
CA track finder: performance and scalability

100 AuAu minimum bias events at 10 AGeV

Efficiency, %	3D	4D 0.1MHz	4D 1MHz	4D 10MHz
All tracks	92.5 %	93.8 %	93.5 %	91.7 %
Primary high-p	98.3 %	98.1 %	97.9 %	96.2 %
Primary low-p	93.9 %	95.4 %	95.5 %	94.3 %
Secondary high-p	90.8 %	94.6 %	93.5 %	90.2 %
Secondary low-p	62.2 %	68.5 %	67.6 %	64.3 %
Clone level	0.6 %	0.6 %	0.6 %	0.6 %
Ghost level	1.8 %	0.6 %	0.6 %	0.6 %
True hits per track	92%	93 %	93 %	93%
Hits per MC track	7.0	7.0	6.97	6.70

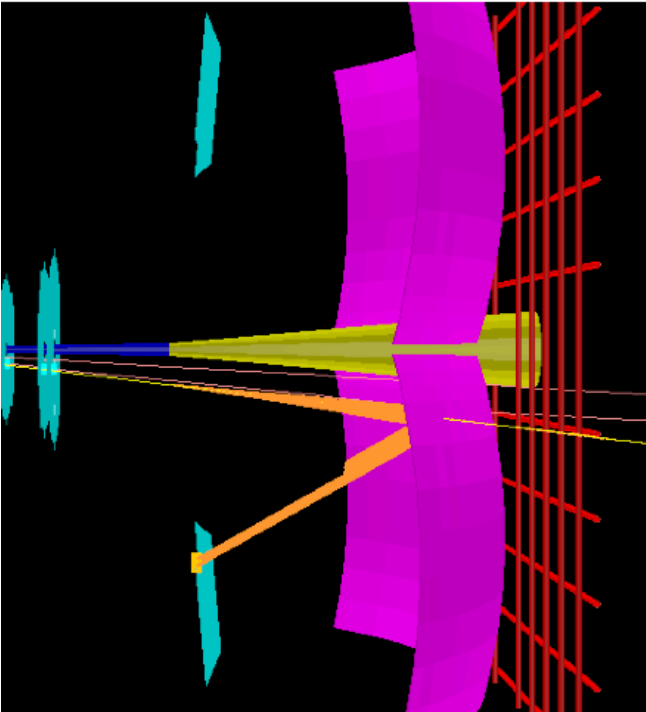
High efficiency for primary tracks

Rate effects become visible above 1 MHz interaction rate



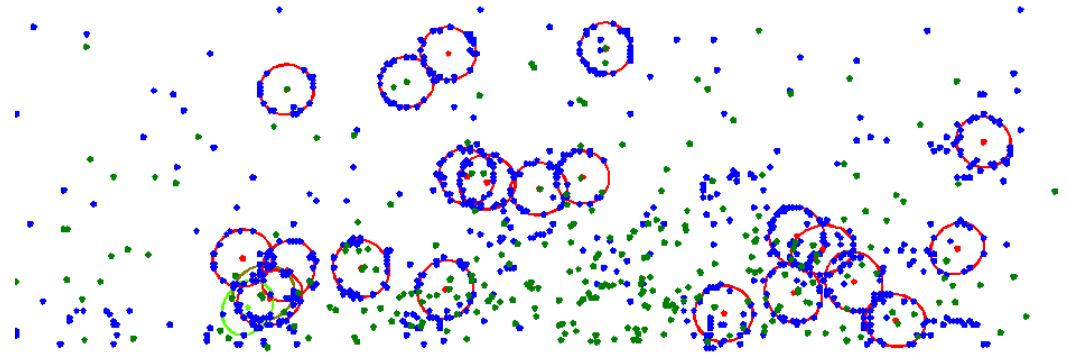
Good scaling behaviour: well suited for many-core systems

Another Example: Ring Finding in the RICH



Cherenkov light emitted by electrons in the radiator is mirrored and focused into rings onto the photodetector plane.

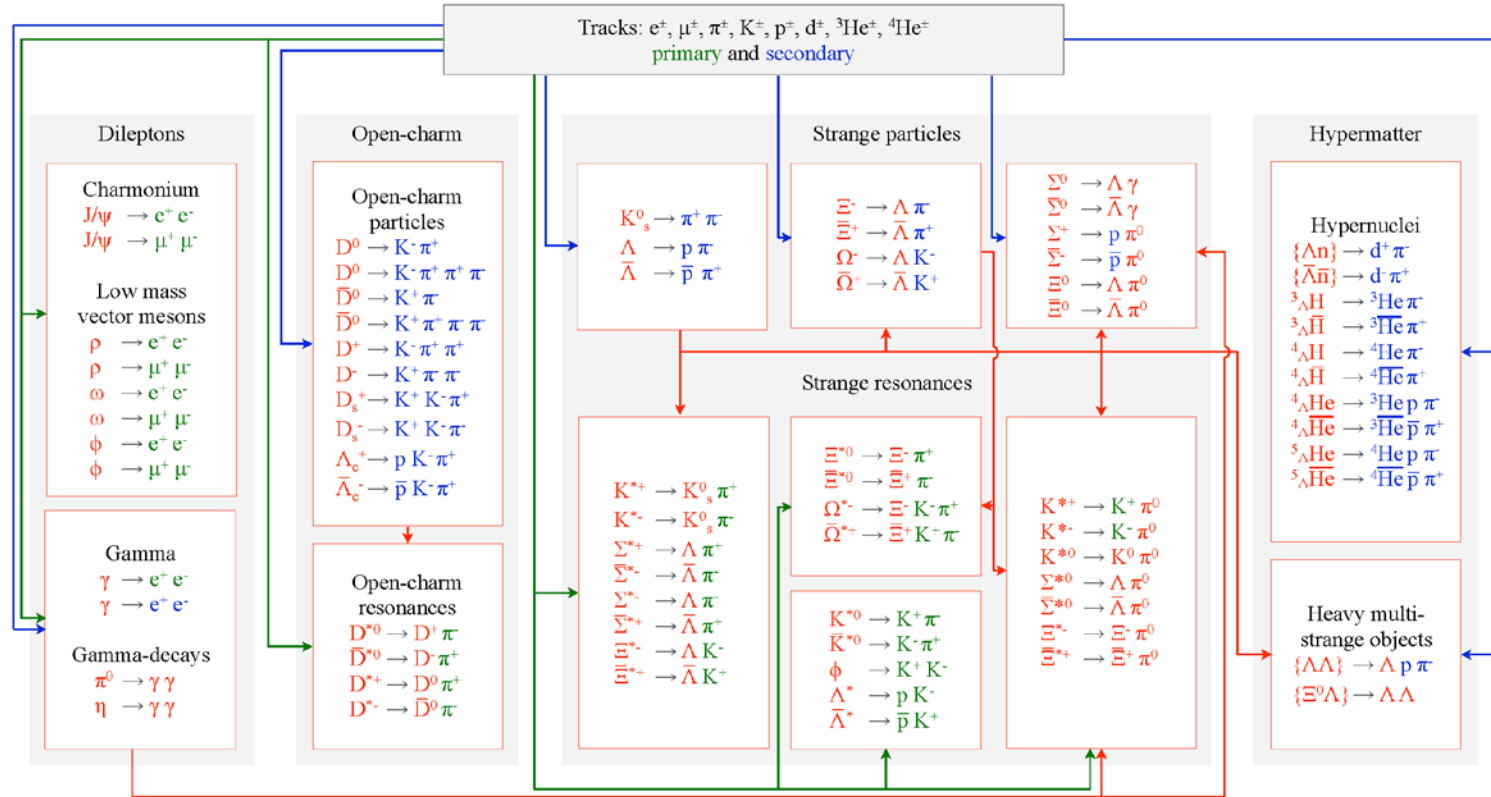
Event Display



Problems:

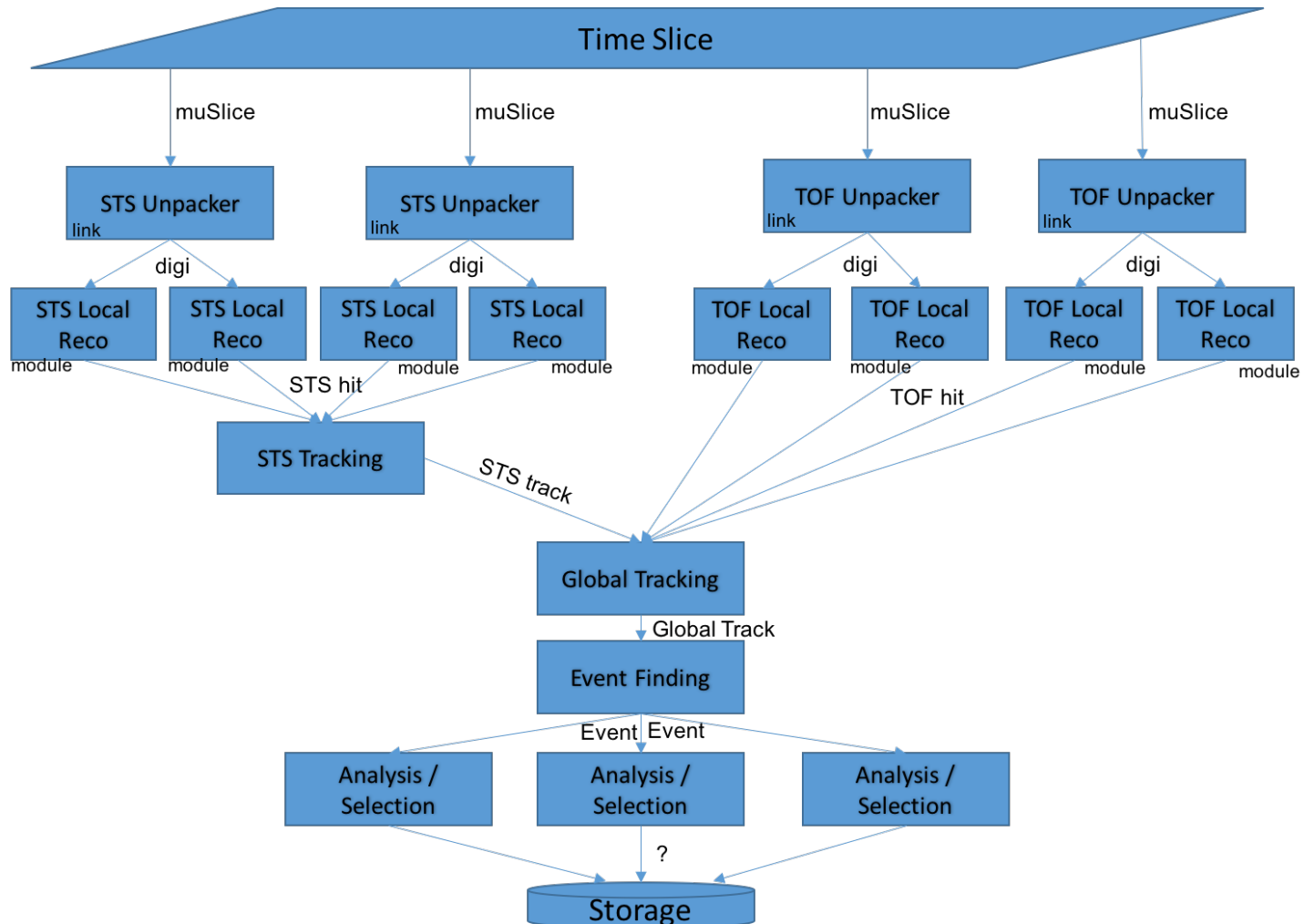
- High hit / ring density
- Overlapping rings
- Ring distortions

Particle Reconstruction in Real-Time



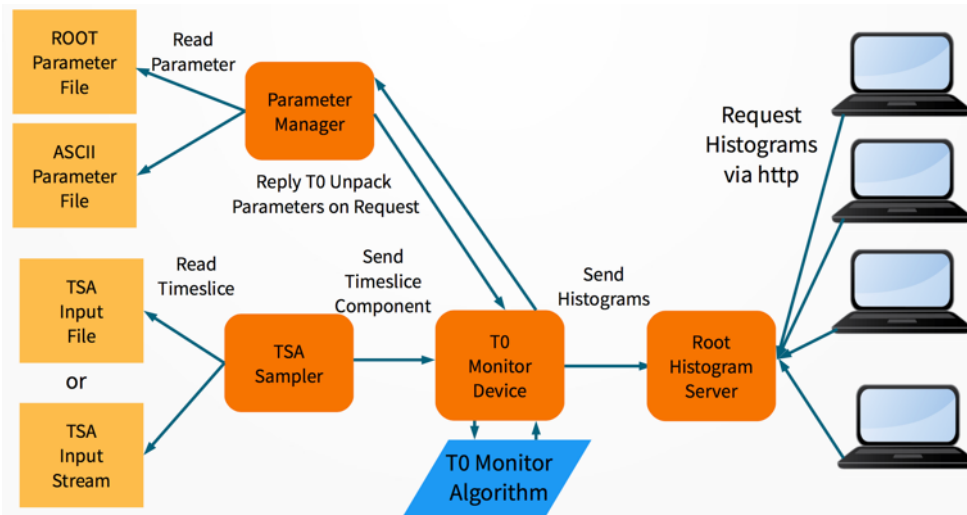
KFParticleFinder: Simultaneous access to multitude of particles
 Real-time reconstruction allows online selection of rare probes.

Example: Simple Process Graph (STS + TOF)



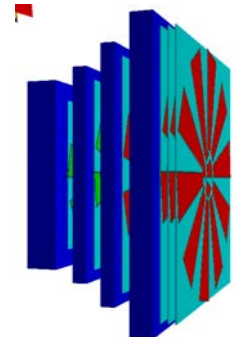
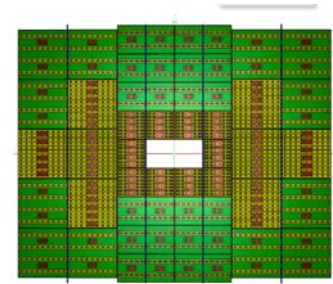
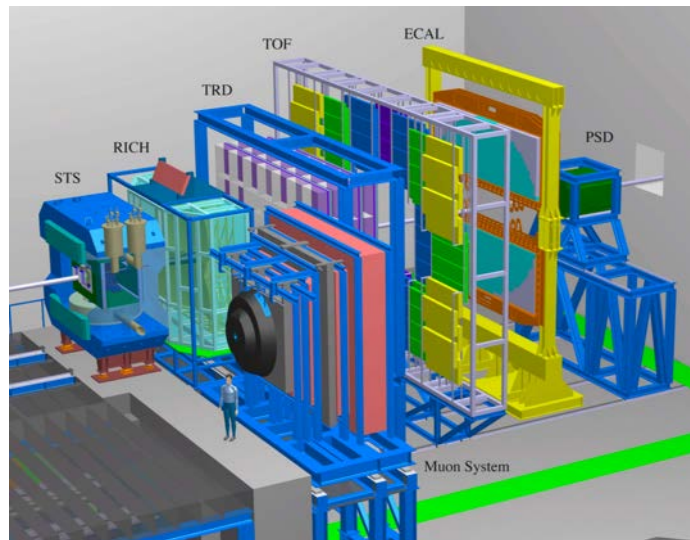
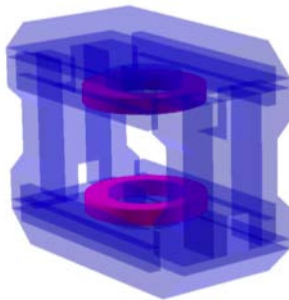
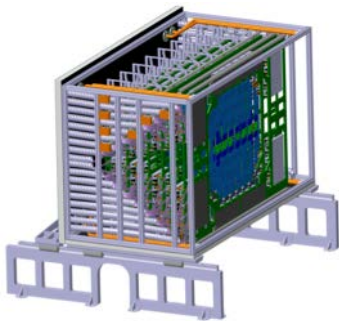
Data Processing Framework

- Shortcoming of the current framework: linear task queue, no concurrency features -> not well suited for online data processing
- Moving to message-queue-based system (FairMQ); intra-node and inter-node data transport possible
- First deployment (proof-of-principle): online monitoring for mCBM

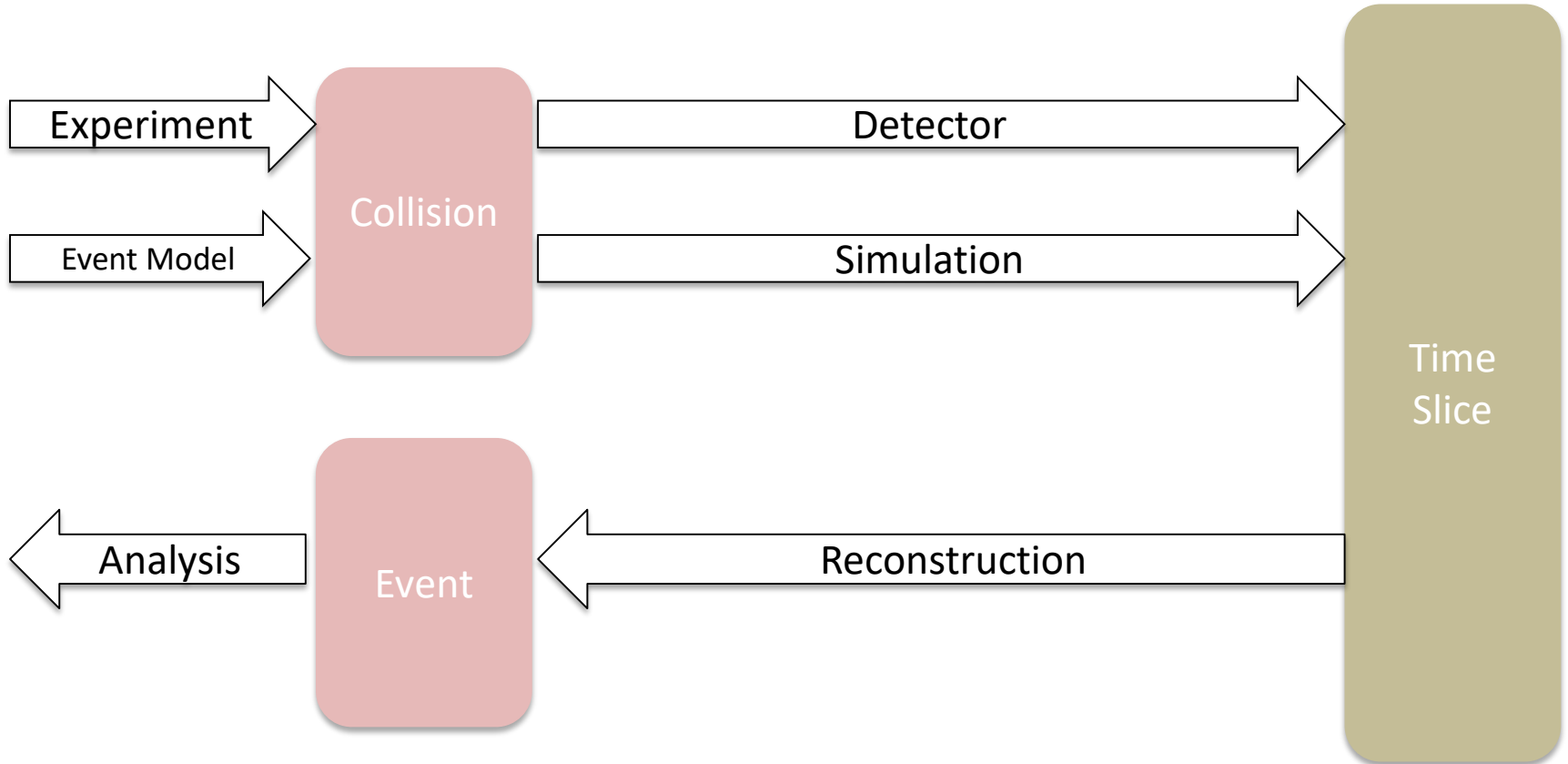


Simulation Software

- Detector geometry model
 - according to current technical planning
 - comprising all relevant contributors to the material budget
 - format: TGeo
 - subject to continuous adjustments / improvements



Events and Time Slices



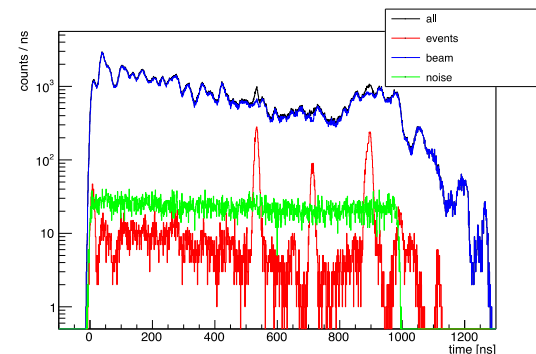
Simulation Software

- Detector response model:
 - analogue response in sensitive detector elements
 - digital response (R/O ASIC): free-streaming
 - model timing response
 - interference between different events
 - thermal noise
- DAQ emulation (time-slice building)
- Goes far beyond conventional event-by-event simulation
 - framework extensions implemented; full data stream can be simulated
 - not yet in real (compressed) raw data format, but logically equivalent
 - combining different sources at different rates (events, beam)

Example: STS

- energy loss fluctuations (Urban model)
- drift to readout surface in bias field
- Lorentz shift
- Thermal diffusion
- Collection on read-out strips
- Cross-talk

Simulated STS data (w/o thermal shielding),
Au+Au @ 10A GeV, beam rate $10^9/s$, event rate $10^7/s$



Summary

- The online computing challenge for CBM (and PANDA) originates from the necessity to be selective w.r.t very rare observable in real-time.
- The offline challenge is to efficiently analyse a huge amount of data by a geographically diverse scientific community.
- Both challenges require the development and deployment of forefront computing technologies.

You Like Challenges?

Then CBM Computing might be of interest for you....welcome!

