Data Acquisition and Trigger of the CBM Experiment

Volker Friese
GSI Darmstadt

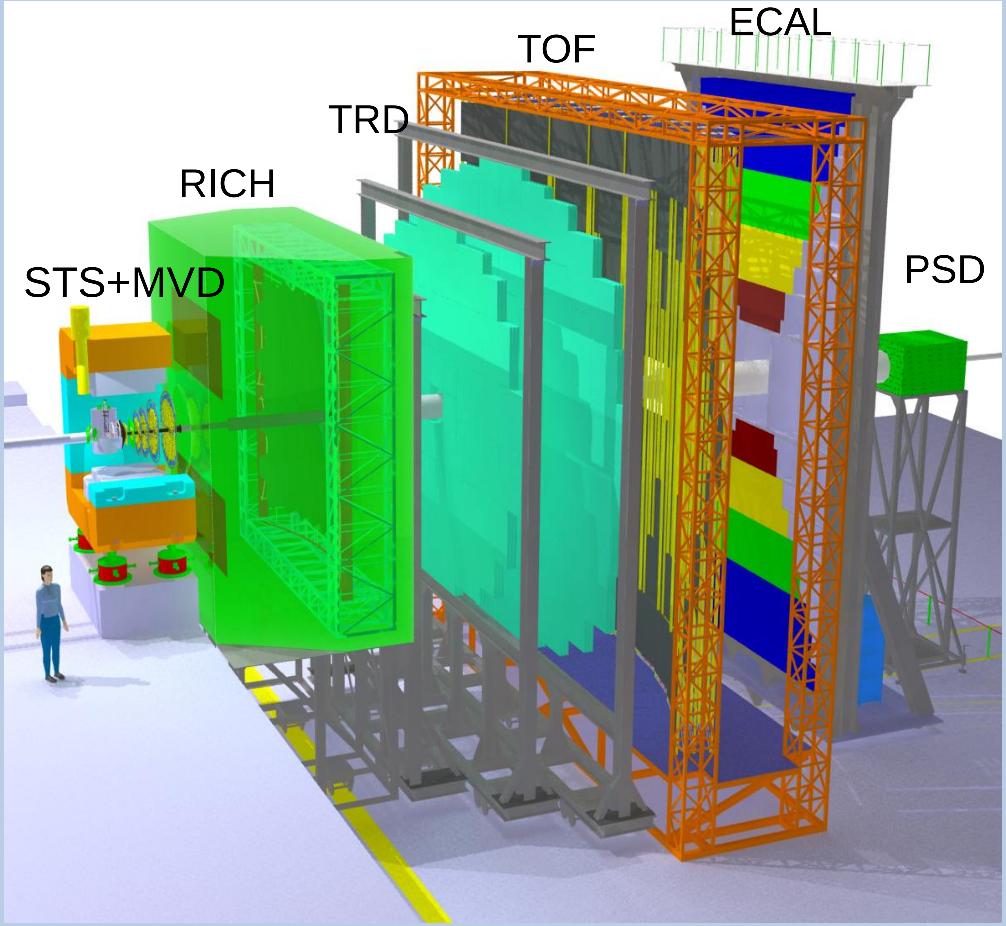
International Conference on Triggering Discoveries in High Energy Physics
11 September 2013, Jammu, India

A very short talk

- Overview of CBM
  - see my presentation of yesterday‘s
- CBM Trigger
  - there will be none
- Thanks for your attention

Reminder: experimental setup

Electron + Hadron setup



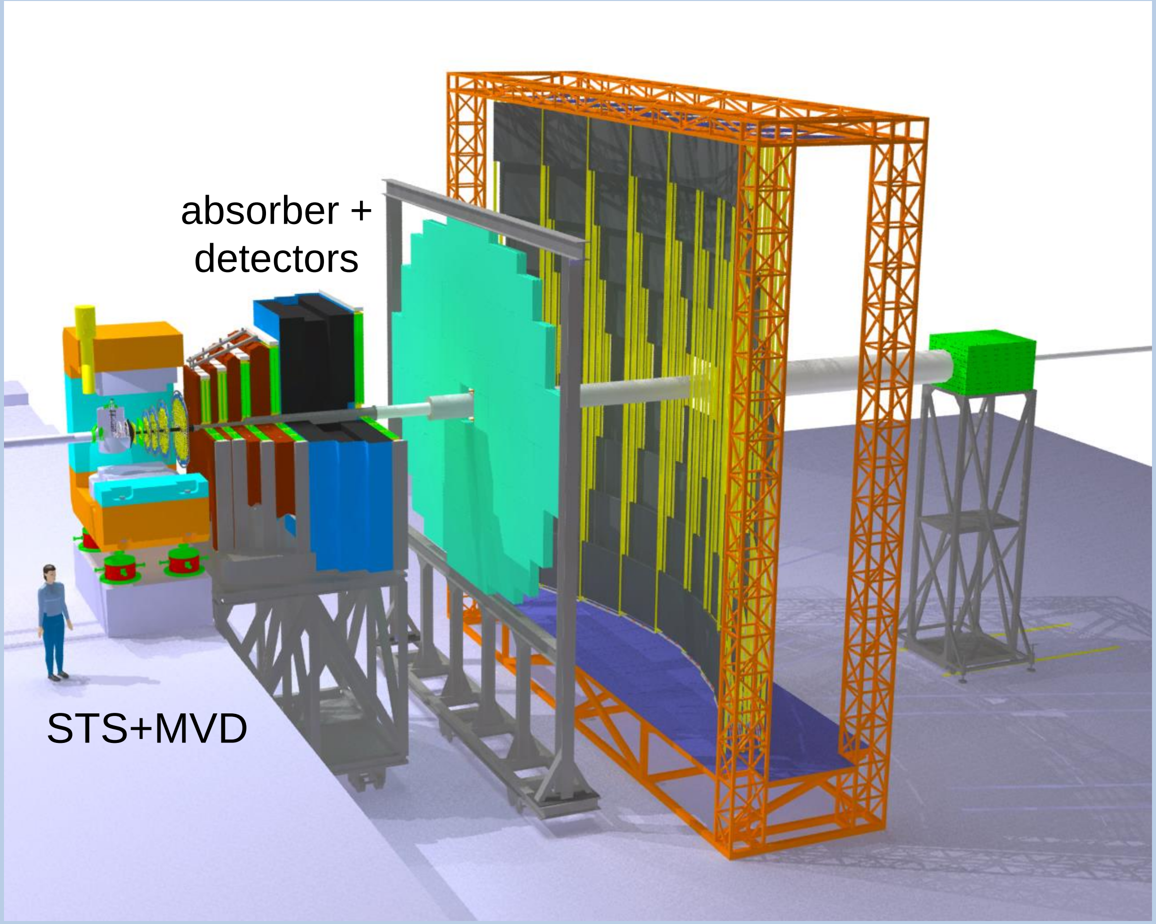
Measurement of hadrons (including open charm) and electrons

Core tracker: STS (silicon strip detectors)

Micro-vertex detector for precision measurement of displaced vertices

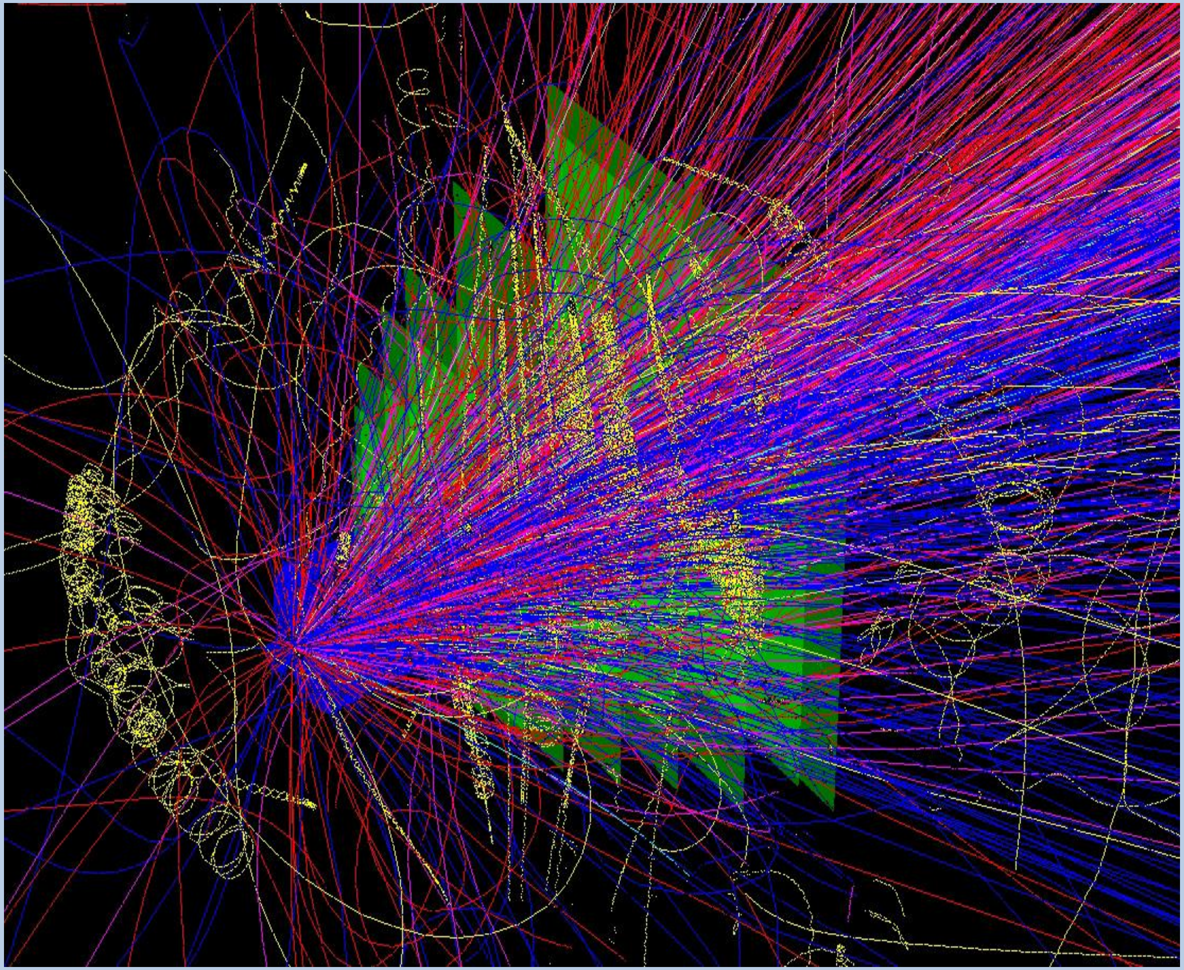
Reminder: experimental setup

Muon setup



Measurement of muons (low-mass and charmonia) in active absorber system

The Challenge



- typical CBM event: about 700 charged tracks in the acceptance
- strong kinematical focusing in the fixed-target setup: high track densities
- up to 10^7 of such events per second
- find very rare signals, e.g., by decay topology, in such a background

...hold it for a second...

10 MHz event rate with heavy ions? You‘re crazy. Past and current experiments run with several Hz to several 100 Hz...

But in particle physics, they have much higher rates.

Yes, but our event topology is much more complex...

So what you think defines the rate limit?

Not in fixed target...

The machine?

You can build fast ones. Just stay away from large drift chambers....

Detectors?

Can also be fast. Just invest a little more and supply proper cooling....

Electronics?

What then?

It‘s the data processing, stupid!

Trigger Considerations

- Signatures vary qualitatively:
  - local and simple: J/ψ->$\mu^+\mu^-$
  - non-local and simple: J/ψ -> e^+e^-
  - non-local and complex: D,Ω->charged hadrons
- For maximal interaction rate, reconstruction in STS is always required (momentum information), but not necessarily of all tracks in STS.
- Trigger architecture must enable
  - variety of trigger patterns (J/ψ: 1% of data, D mesons: 50% of data)
  - multiple triggers at a time
  - multiple trigger steps with subsequent data reduction
- Complex signatures involve secondary decay vertices; difficult to implement in hardware.
- Extreme event rates set strong limits to trigger latency.

Running Conditions

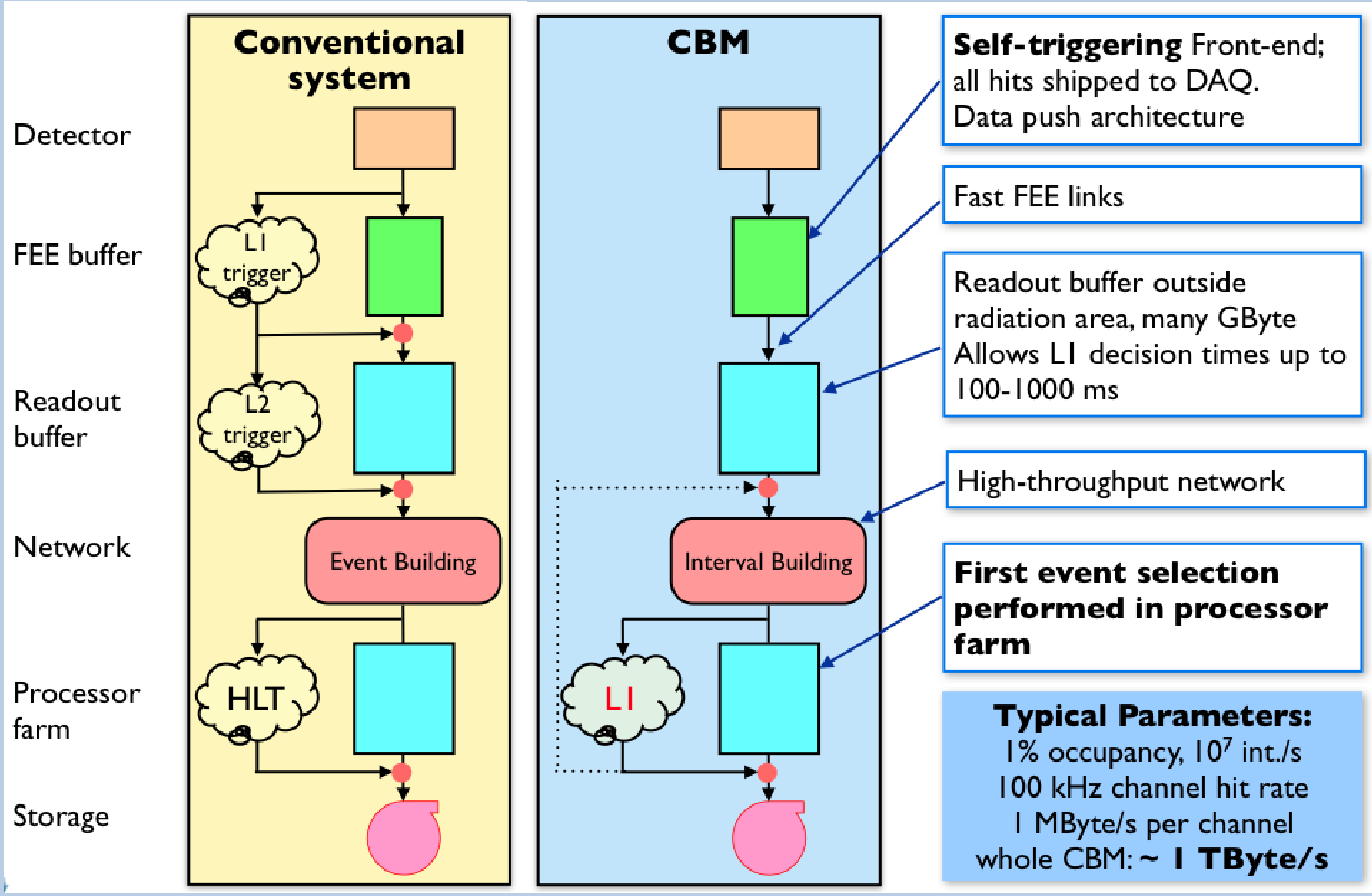
Condition	Interaction rate	limited by	Application
No Trigger	10^4/s	archival rate	bulk hadrons, low-mass di-electrons
Medium Trigger	10^5 /s – 10^6/s	MVD (speed, rad. tolerance), trigger signature	open charm multi-strange hyperons, low-mass di-muons
Max. Trigger	- 10^7/s (even more for p beam)	on-line event selection	charmonium

Detector, FEE and DAQ requirements are given by the most extreme case

Design goal: 10 MHz minimum bias interaction rate

Requires on-line data reduction by up to 1,000

CBM Readout Concept



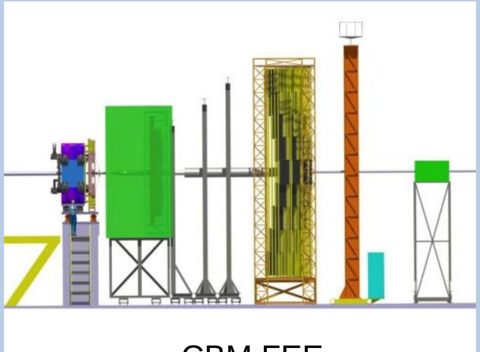
Finite-size FEE buffer: latency limited

throughput limited

Consequences

- The system is limited only by the throughput capacity and by the rejection power of the online computing farm.
- There is no a-priori event definition: data from all detectors come asynchroneously; events may overlap in time.
- The classical DAQ task of „event building“ is now rather a „time-slice building“. Physical events are defined later in software.
- Data reduction is shifted entirely to software: maximum flexibility w.r.t. physics

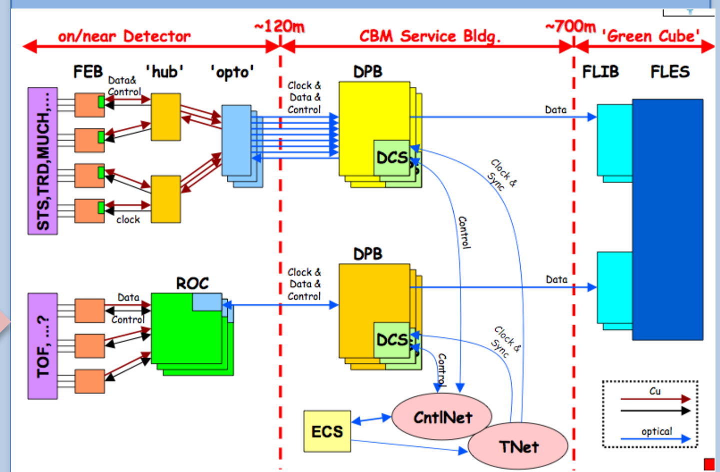
The Online Task



CBM FEE

1 TB/s

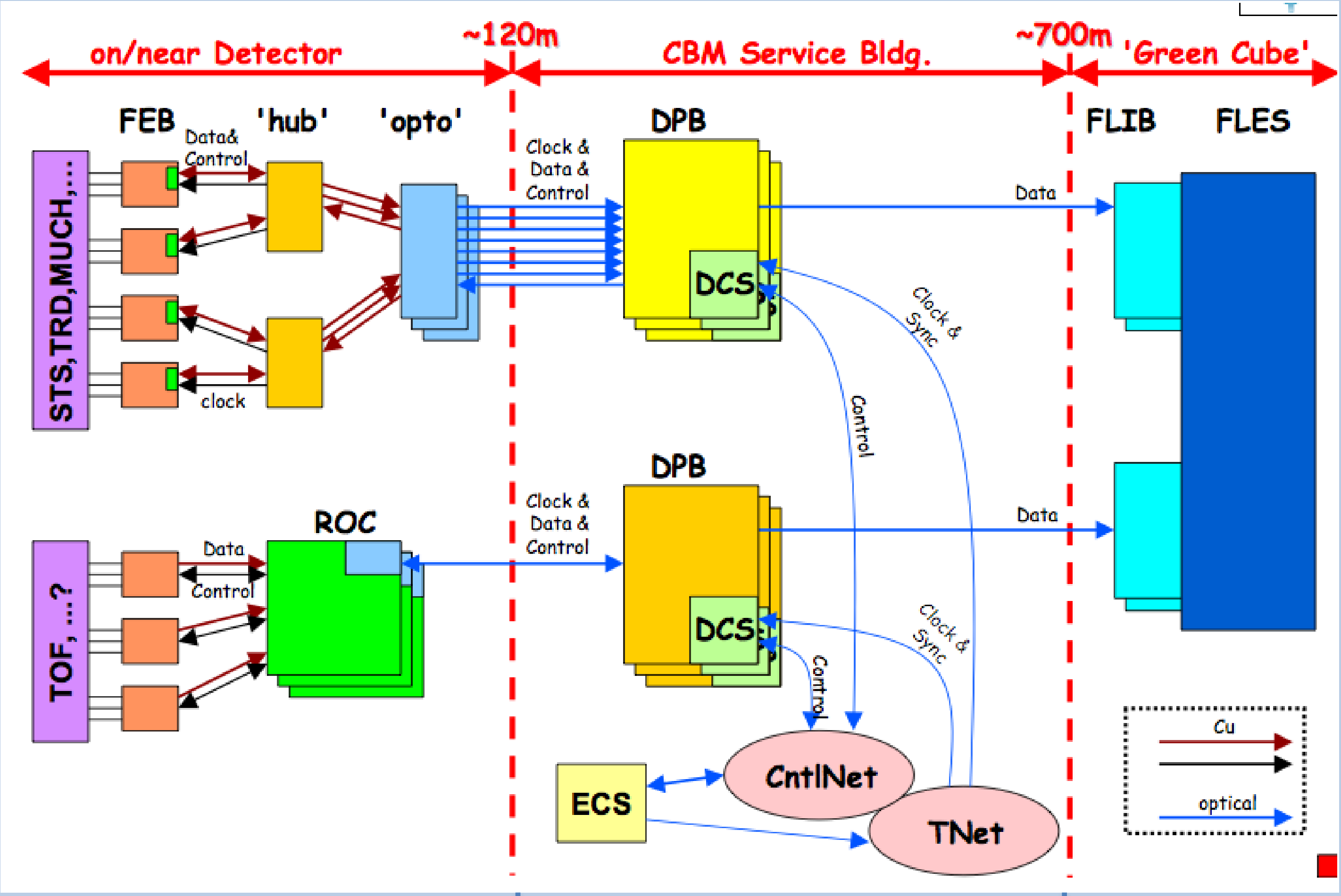
at max. interaction rate



1 GB/s

Mass Storage

CBM Readout Architecture

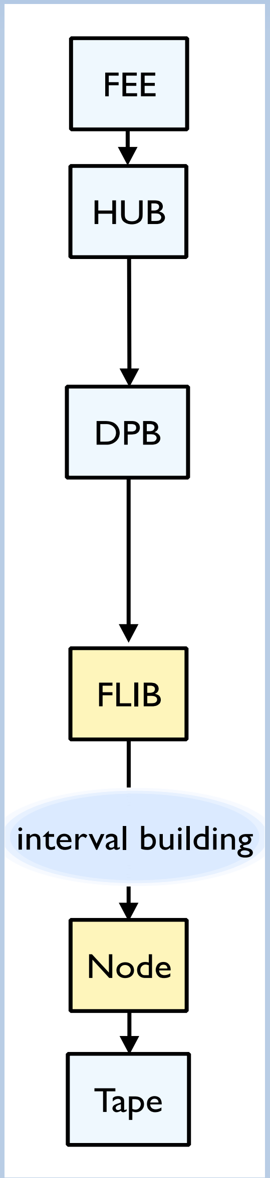


DAQ: data aggregration time-slice building (pre-processing?)

FLES: event reconstruction and selection

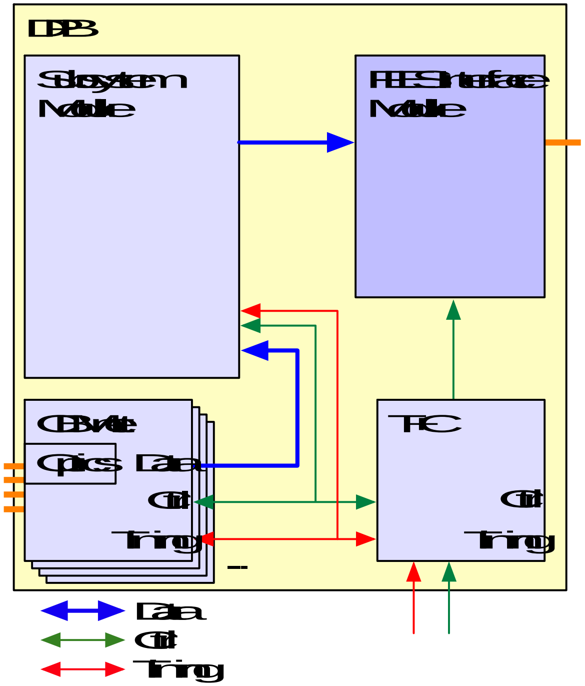
Components of the read-out chain

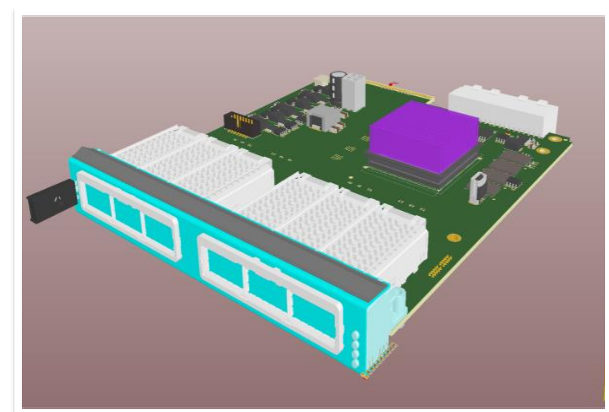
- Detector Front-Ends
  - each channel performs autonomous hit detection and zero suppression
  - associate absolute time stamp with hit, aggregate data
  - data push architecture
- Data Processing Board (DPB)
  - perform channel and segment local data processing
    - feature extraction, time sorting, data reformatting , merging input streams
  - time conversion and creation of microslice containers
- FLES Interface Board (FLIB)
  - time indexing and buffering of microslice containers
  - data sent to FLES is concise: no need for additional processing before interval building
- FLES Computing Nodes
  - calibration and global feature extraction
  - full event reconstruction (4-d)
  - event selection



Data Processing Board

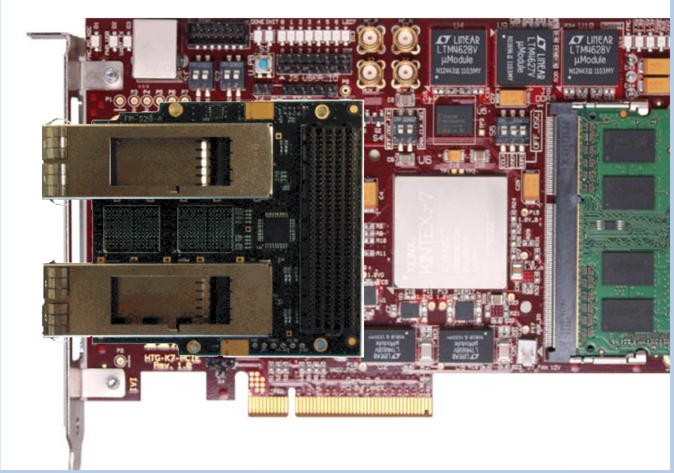
- FPGA-based concentrator and processor board
- Located in the CBM Service Building
- Interfaces all subsystems:
  - Unified optical link to detector FEE components
  - Link to FLES (long distance)
  - DCS (control, clock and sync)
- Subsystem specific data processing
- Coordinate front-end
  - System synchronization
  - FEE control, throttling
- Build microslice containers
  - Partition data stream
  - Add status information as required
- Can provide FLES-less readout for test purposes
- MTCA based DPB layer currently under development



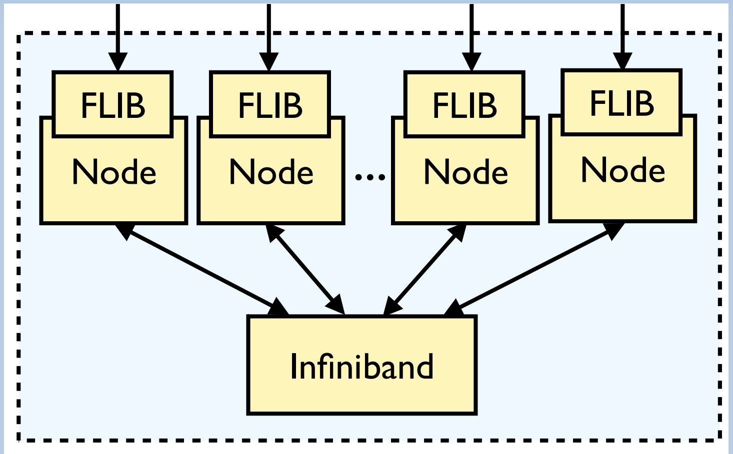


FLES Interface Board (FLIB)

- PCIe add-on board to connect FLES nodes and DPB
- Tasks:
  - consumes microslice containers received fro DPB
  - time indexing of MC for interval building
  - transfer MCs and index to PC memory
- Current development version:
  - test platform for FLES hardware and software developments
  - readout device for testbeams and lab setups
- Requirements:
  - fast PCIe interface to PC
  - high number of optocal links
  - large buffer memory
- Readout firmware for Kintex-7 based board under development

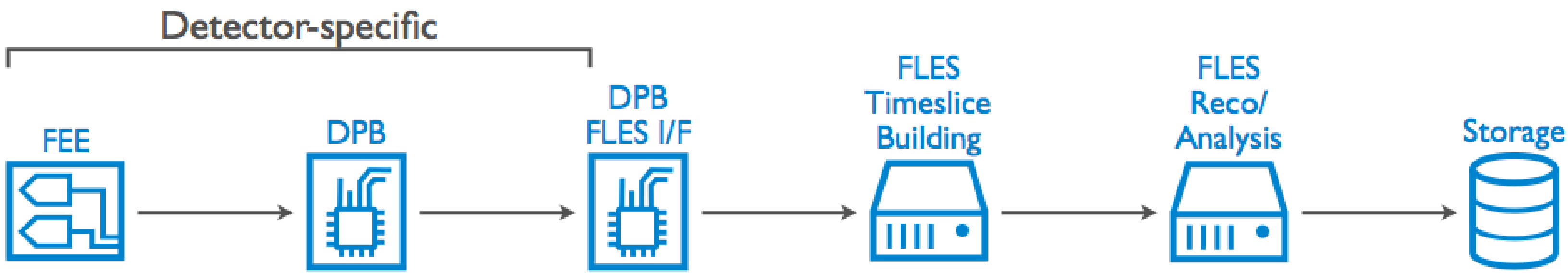


FLES Architecture



- FLES is designed as HPC cluster
  - commodity hardware
  - GPGPU accelerators
- Total input rate ~1 TB/s
- Infiniband network for interval building
  - high throughput, low latency
  - RDMA dara transfer, convenient for interval building
  - most-used system interconnect in latest top-50 HPC
- Flat structure; input nodes distributed over the cluster
  - full use of Infiniband bandwidth
  - input data is concise, no need for processing bevor interval building
- Decision on actual hardware components as late as possible

Data Formats



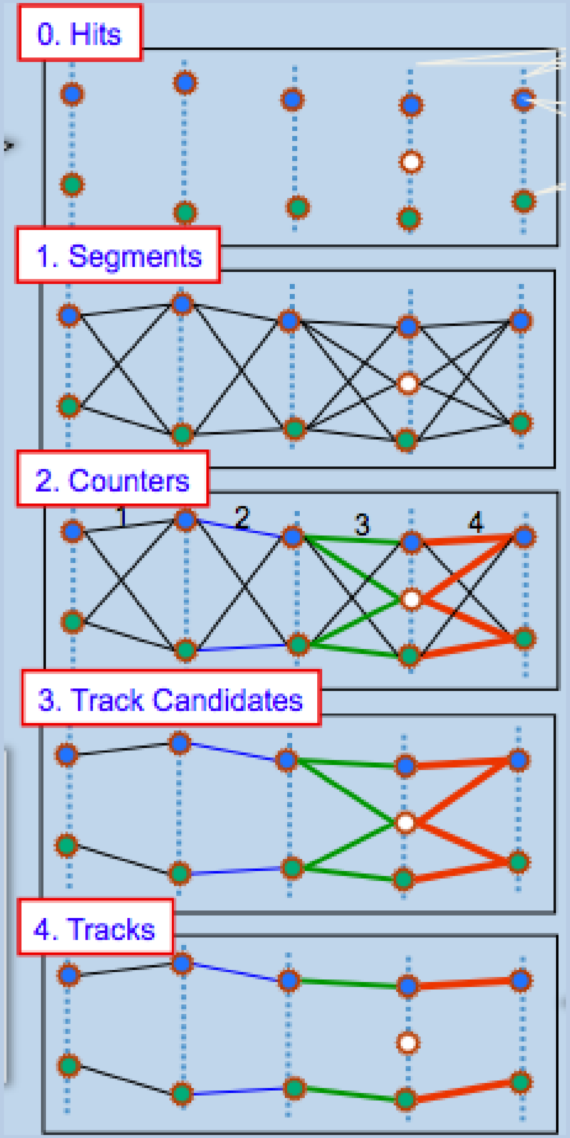
Detector **Message** Format	Detector **Data** Format	Microslice Containers	Timeslice Containers	Storage Data
• Data points, epoch markers, etc. • Very small fundamental messages • Mostly specified by FEE ASICS	• Detector data for a constant time interval • Preprocessed or raw data • Self-contained • Data contents of an MC	• Lightweight container format • 128-bit header • Preliminary specification available	• Timeslices for analysis • Concatenation of microslice containers • Index table	• Data of selected events • Possibly ROOT files

FLES location

Online reconstruction and data selection

- Decision on interesting data requires (partial) reconstruction of events:
  - track finding
  - secondary vertex finding
  - further reduction by PID
- Throughput depends on
  - capacity of online computing cluster
  - performance of algorithms
- Algorithms must be fully optimised w.r.t. speed, which includes full parallelisation
  - tailored to specific hardware (many-core CPU, GPU)
  - beyond scope of common physicist; requires software experts

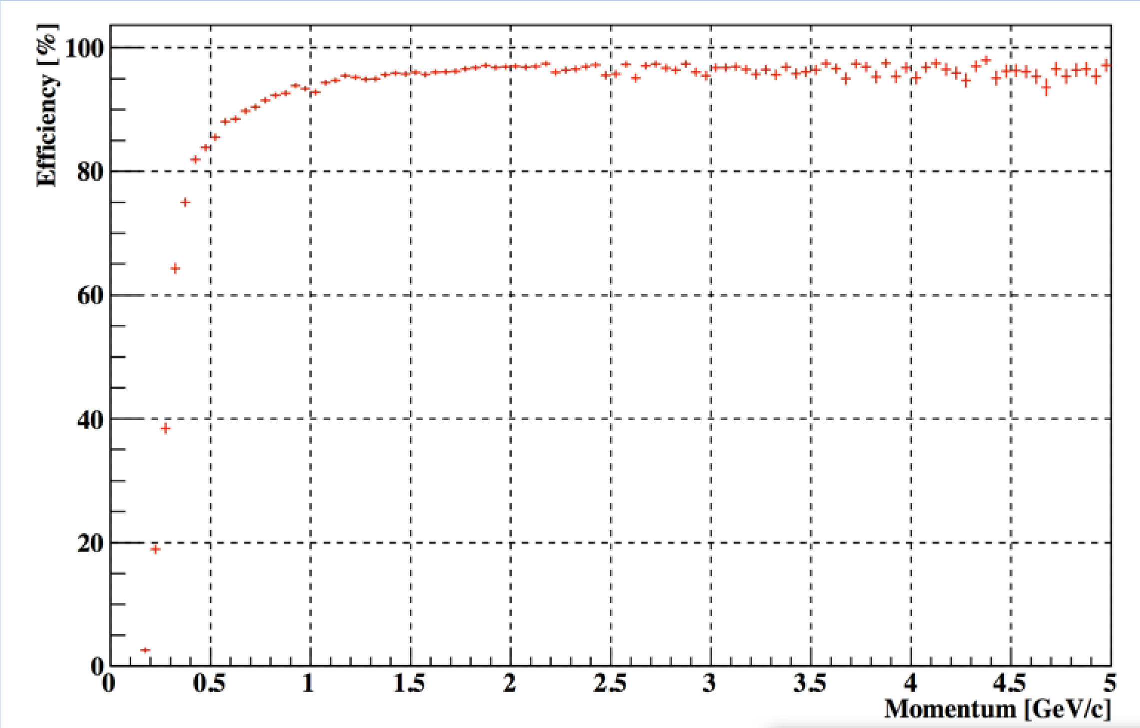
Reconstruction backbone: Cellular Automaton in STS



- cells: track segments based on track model
- find and connect neighbouring cells (potentially belonging to the same track)
- select tracks from candidates

- simple and generic
- efficient and very fast
- local w.r.t. data and intrinsically parallel

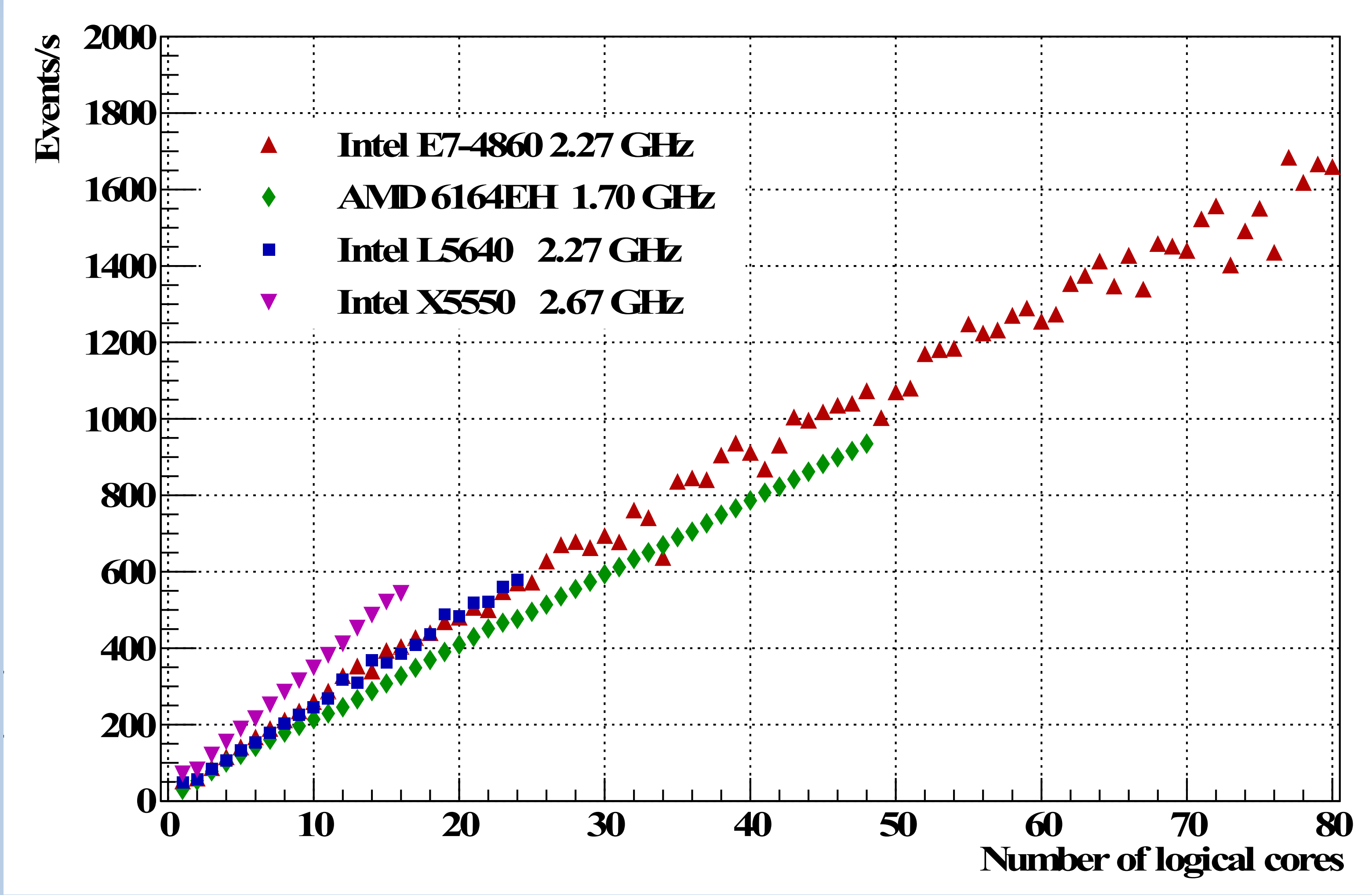
CA performance



	Efficiency, %	
	mbias	central
Primary high-*p* tracks	97.1	96.2
Primary low-*p* tracks	90.4	90.7
Secondary high-*p* tracks	81.2	81.4
Secondary low-*p* tracks	51.1	50.6
All tracks	88.5	88.3
Clone level	0.2	0.2
Ghost level	0.7	1.5
Reconstructed tracks/event	120	591
Time/event/core	8.2 ms	57 ms

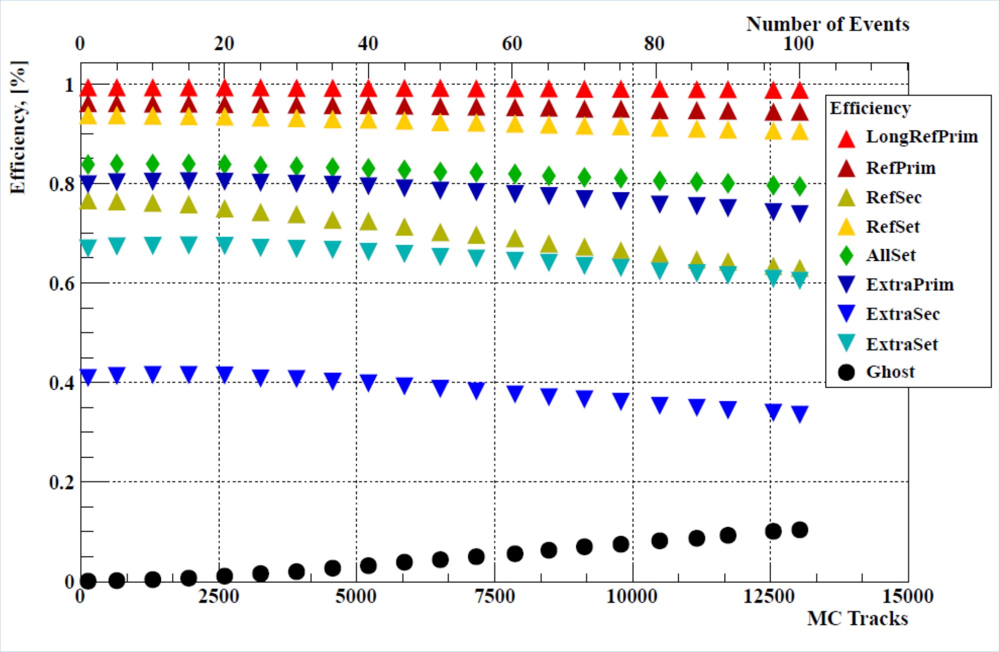
STS track finding with high efficiency on 10 ms level

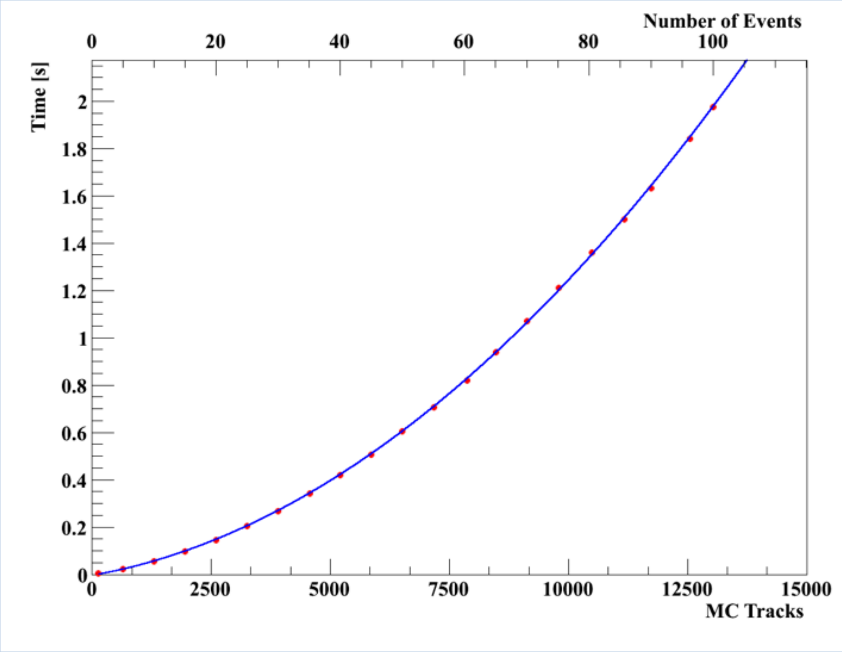
CA scalability



Good scaling beviour: well suited for many-core systems

CA stability





Stable performance also for large event pile-up

Many more tasks for online computing

- Track finding in STS
- Track fit
- Track finding in TRD
- Track finding in Muon System
- Ring finding in RICH
- Matching RICH ring, TOF hit and ECAL cluster to tracks
- Vertexing
- Analysis and data selection

Parallelisation in CBM reconstruction

Algorithm	Vector SIMD	MultiThreading	CUDA	OpenCL CPU/GPU
Hit Producers				
STS KF Track Fit	✓	✓	✓	✓/✓
STS CA Track Finder	✓	✓		
MuCh Track Finder	✓	✓	✓	
TRD Track Finder	✓	✓	✓	
RICH Ring Finder	✓	✓		(✓/✓)
Vertexing (KFParticle)	✓	✓		
Off-line Physics Analysis	✓			
FLES Analysis and Selection	✓	✓		

Andrzej Nowak (OpenLab, CERN) by Hans von der Schmitt (ATLAS) at GPU Workshop, DESY, 15-16 April 2013

	SIMD	Instr. Level Parallelism	HW Threads	Cores	Sockets	Factor	Efficiency
MAX	4	4	1.35	8	4	691.2	100.0%
Typical	2.5	1.43	1.25	8	2	71.5	10.3%
HEP	1	0.80	1	6	2	9.6	1.4%
CBM@FAIR	4	3	1.3	8	4	499.2	72.2%

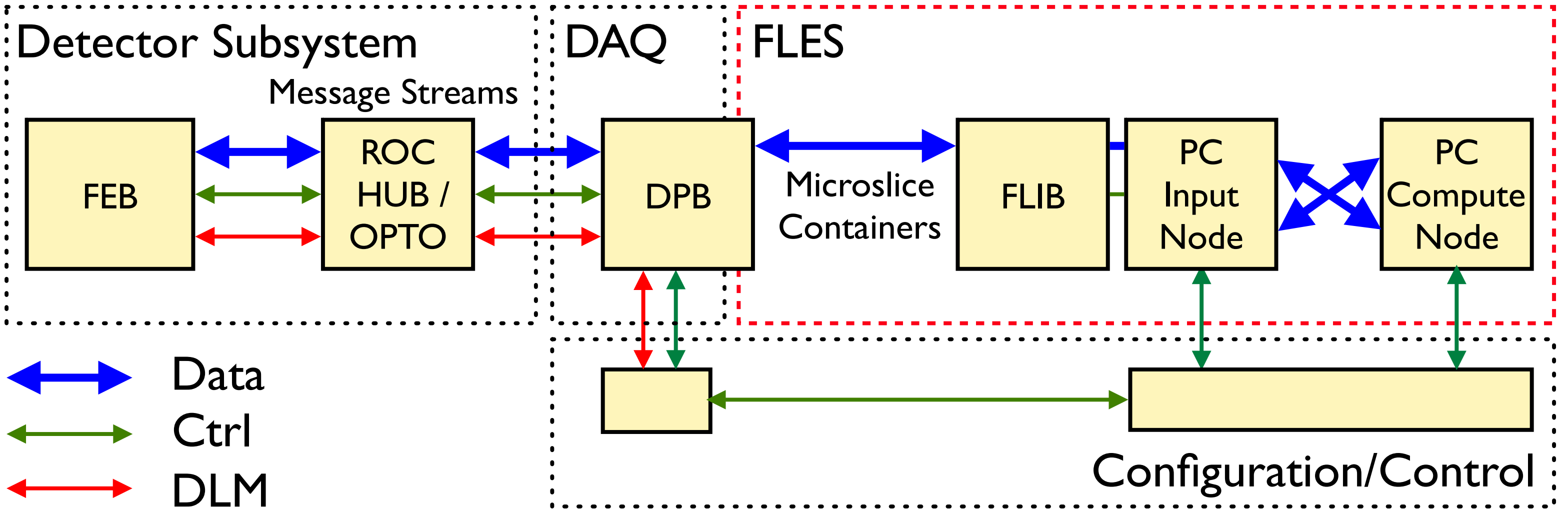
Summary

- CBM will employ no hardware trigger.
- Self-triggered FEE will ship time-stamped data as they come to DAQ.
- DAQ aggregrates data and pushes them to the FLES.
- Transport containers are micro slices and timeslices.
- Online reconstruction and data selection will be done in software on the FLES (HPC cluster).
- Fast algorithms for track finding and fitting have been developed; parallelisation and optimisation of entire reconstruction chain is in good progress.

Material provided by J. de Cuveland, D. Hutter, I. Kisel, I. Kulakov and W. Müller. Thanks!

Backup

Introducing Microslices



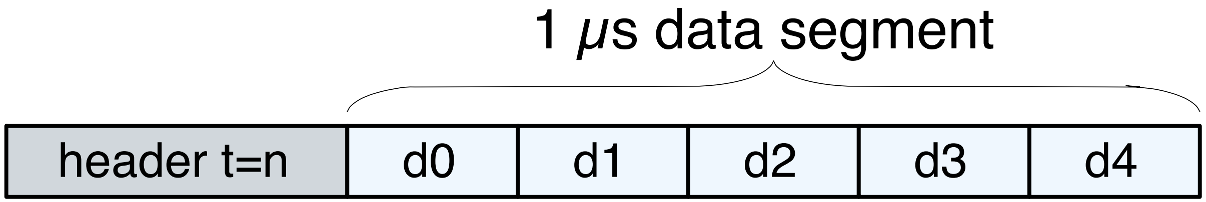
Motivation

- FLES needs to build global intervals to enable reconstruction
- Detector data streams...
  - have to be analyzed w.r.t. time information
  - have to be partitioned (without data loss)
- But: no global time in data stream, stream format subsystem-specific
- A mechanism for interval overlap and two-staged interval building is needed

Solution

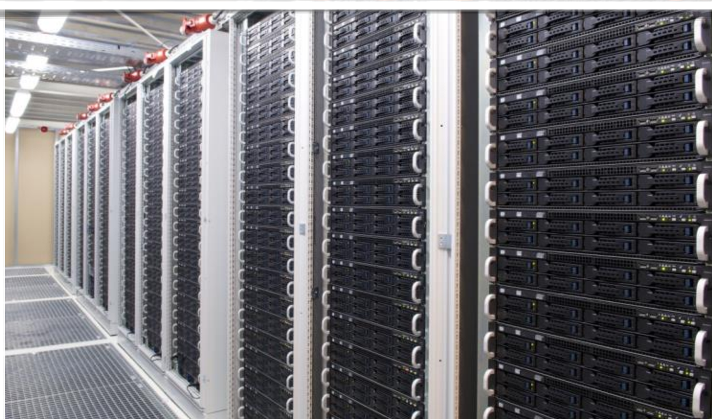
- Partition data streams into „microslices containers (MC)“
- Use detector-specific DPB design to build MCs
- Base FLES timeslice building only on MCs

Microslice-based Interval Building



- MC are constant in time and variable in data size
- Each MC consists of a header and a data segment
  - Header contains start time of corresponding data segment and all other information needed for interval building
  - Data segment contains self-contained subsystem data, meaning it is stateless and does not depend on any previous or following MC
- FLES uses time information from MC for interval building
- Subsequent MC get combined to one processing interval
  - To address interval overlap MC are doubled at the interval end
  - Single MC are addressable for two-staged interval building
- Assumption: each MC is ~1μs in experiment-time (~ 1kB average data size for full link utilization)

MicroFLES Setup at GSI Minicube, First Floor



- 8+1 nodes
  - 100 CPU cores (Intel E5-2620)
  - Dual-processor/NUMA system
  - 544 GB RAM total
  - PCIe Gen 3.0: 16x slots for 1 FLIB + up to 3 GPUs (not yet) per node
- InfiniBand FDR network
  - Managed switch
  - >100 GBit/s IB bandwidth per node
- Status
  - Installed at GSI Minicube in Testing Hall
  - Running reliably for 7 month