Trigger-less readout at LHCb

The 2021 International Workshop on the High Energy Circular Electron Positron Collider

Flavio Pisani for the LHCb Online team

CERN

Nov 08 - 12, 2021

Outline

Introduction to the LHCb experiment and DAQ for Run3

Hardware requirements

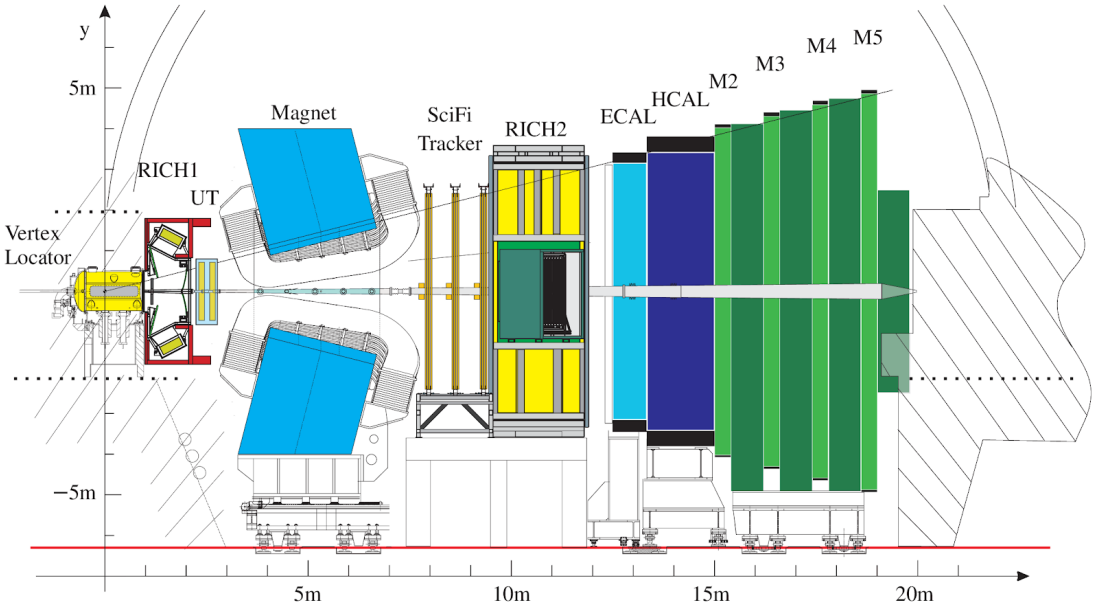
Event building at LHCb

Event building benchmarks and commissioning

Data processing and event selection

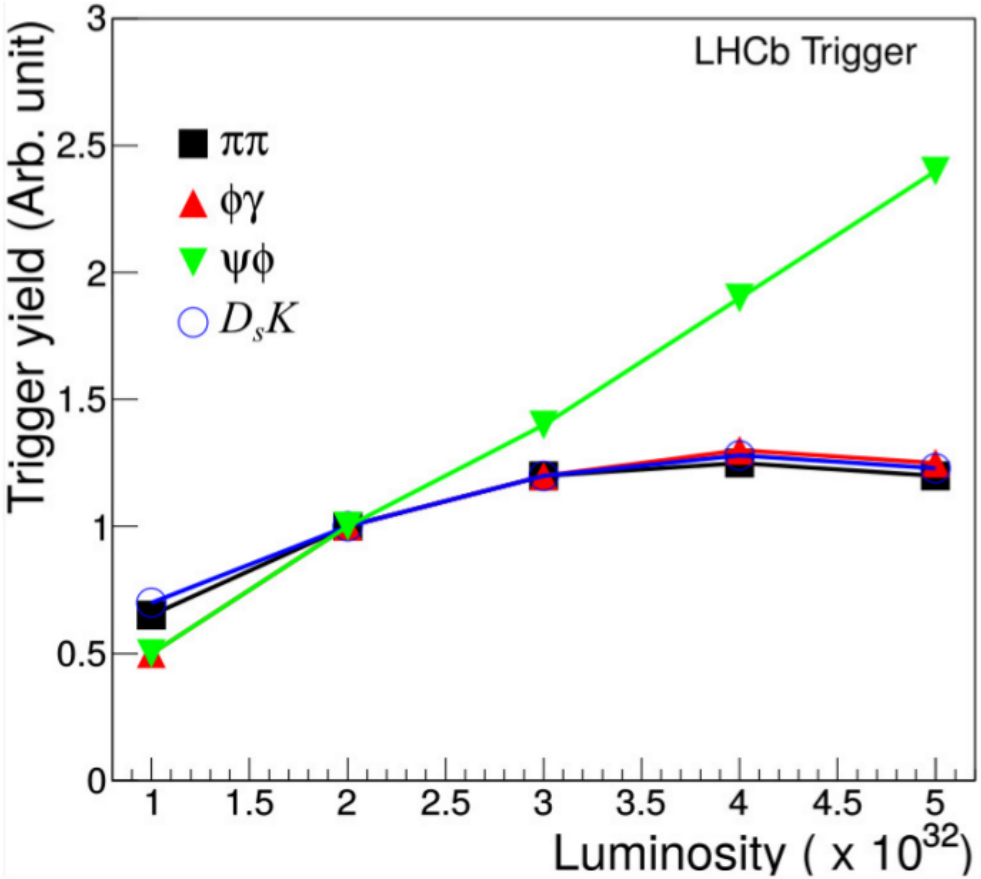
The LHCb experiment

- Forward spectrometer
- Optimised for heavy flavour Physics
- Tracking system:
  - VErtex LOcator (VELO)
  - Upstream Tracker (UT)
  - Scintillating Fibre (SciFi)
- Particle Identification:
  - Ring Imaging Cherenkov (RICH)
  - Calorimeters
  - Muon system



Why a triggerless readout?

Low level trigger yield vs Luminosity ($cm^{-2}s^{-1}$)

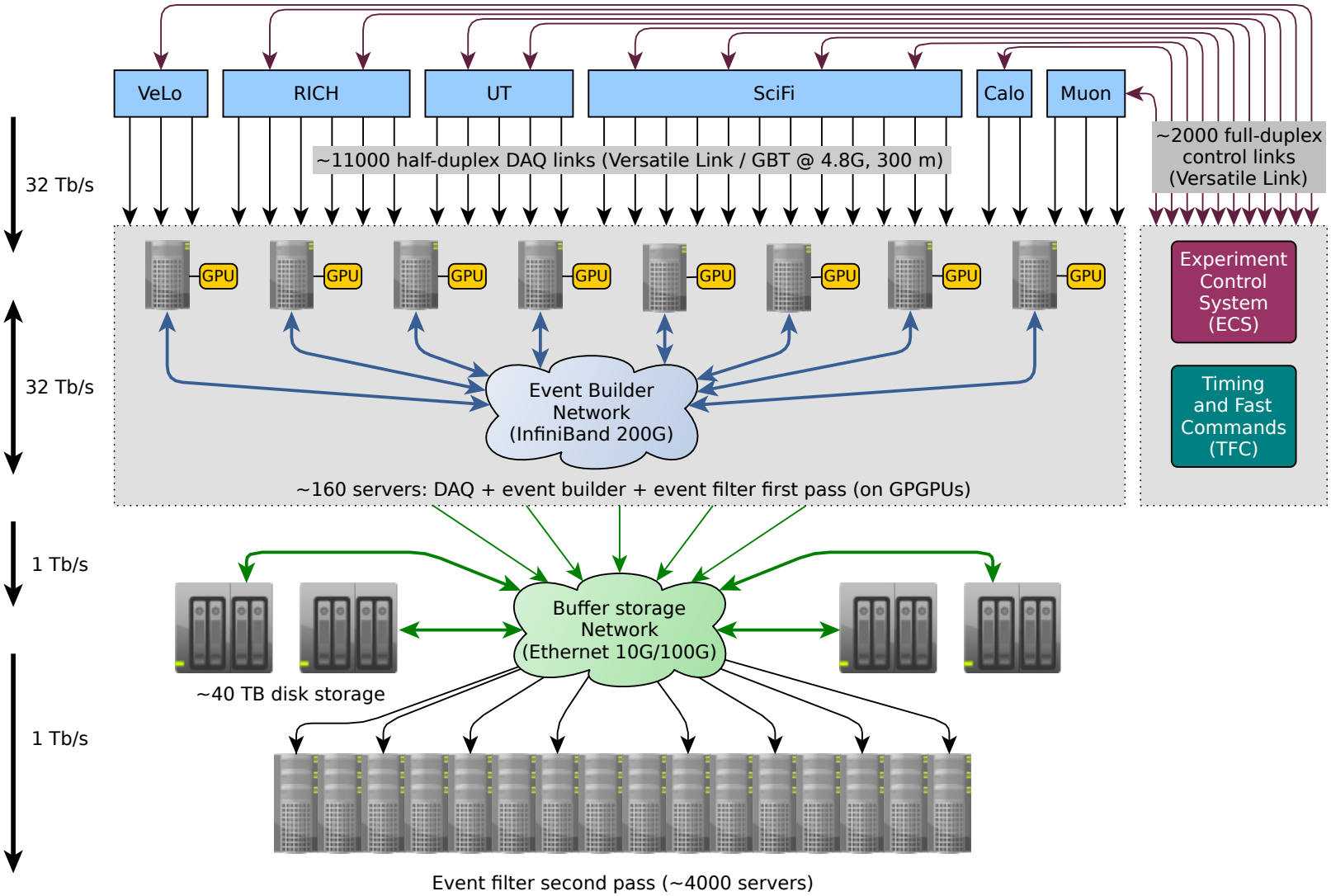


The low level trigger is not efficient at high luminosity

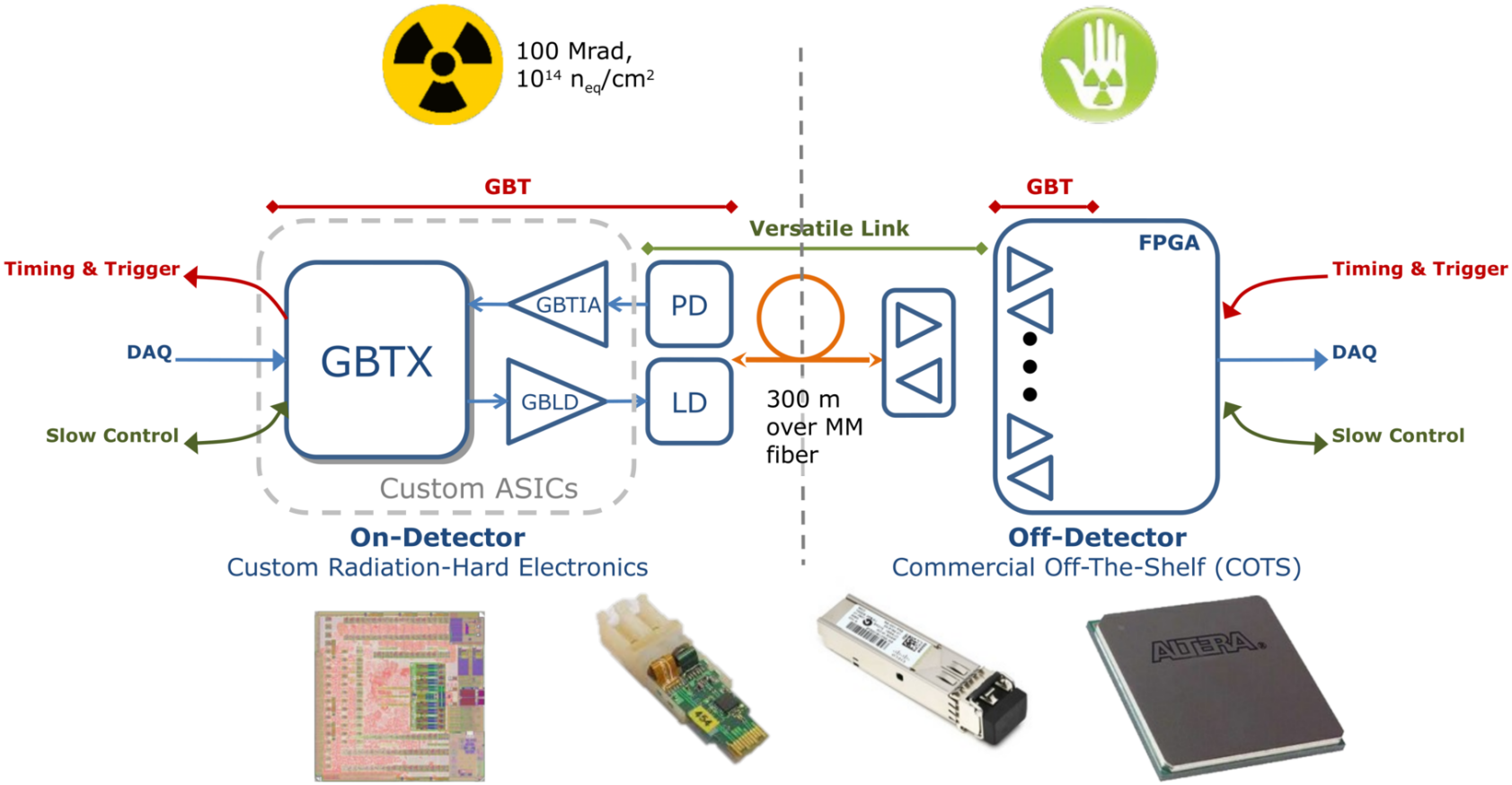
Why can we read out every collision?

- Spectrometer geometry: fibres/cables are not "in the way"
- Relatively low radiation levels permit to relax the constraint on the FPGAs used for "middle" layer processing
- "Zero-suppression" on the detectors
- Total event-size comparatively small ($\sim$ 100 kB)
- Software trigger can do online selection with offline-like reconstruction

LHCb Online system



Front-end connection: GBT over Versatile Link



Credit: P. Moreira, S. Baron (CERN)

The PCIe40: a single custom-made FPGA board for DAQ and Control

- Intel Arria10 FPGA
- 48x10G capable transceiver on 8xMPO for up to 48 full-duplex Versatile Links
- 2 dedicated 10G SFP+ for timing distribution
- 2x8 Gen3 PCIe

One board / many firmware personalities

1 Readout Supervisor (SODIN):

- Reception and distribution of global timing
- Generation and distribution of synchronous and asynchronous command
- Generation of events veto, triggers and calibration events

One board / many firmware personalities

42 Interface Boards (SOL40):

- Distribution of the global timing to the front-end cards
- Interface bridge between the control system and the front-ends

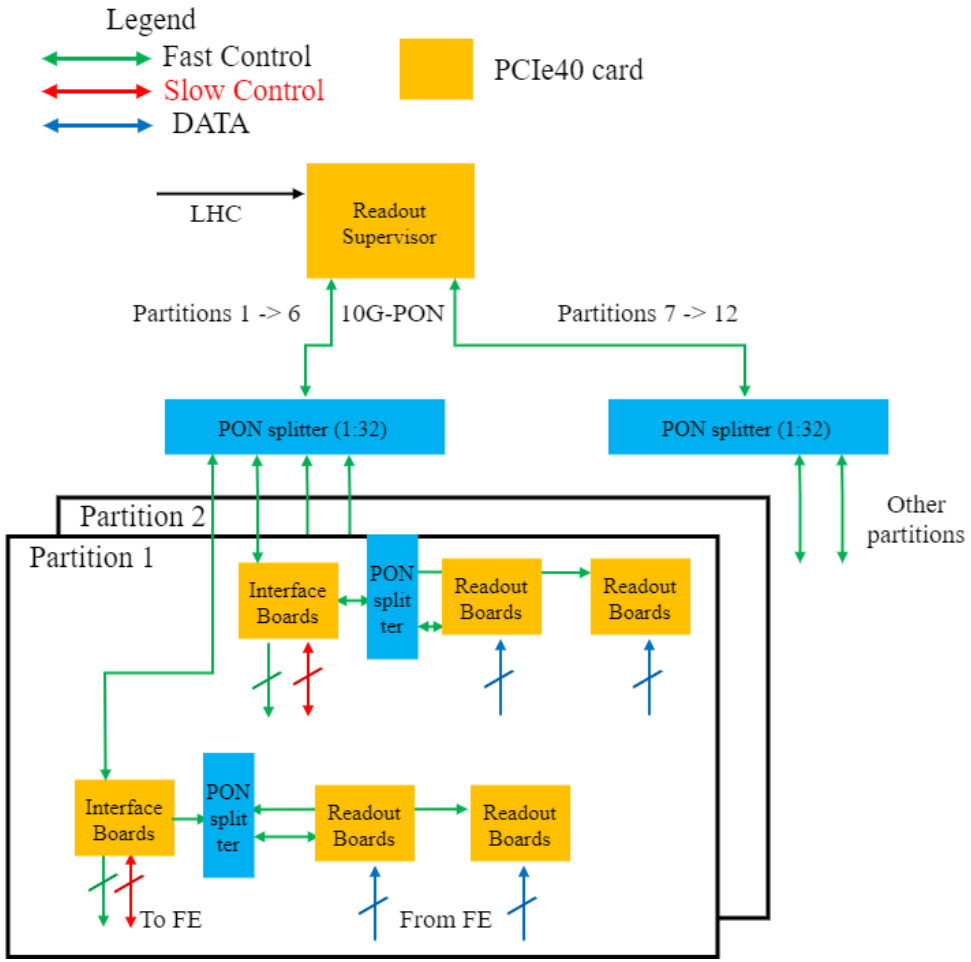
One board / many firmware personalities

478 Readout Boards TELL40:

- Data acquisition
- First pre-processing of the data

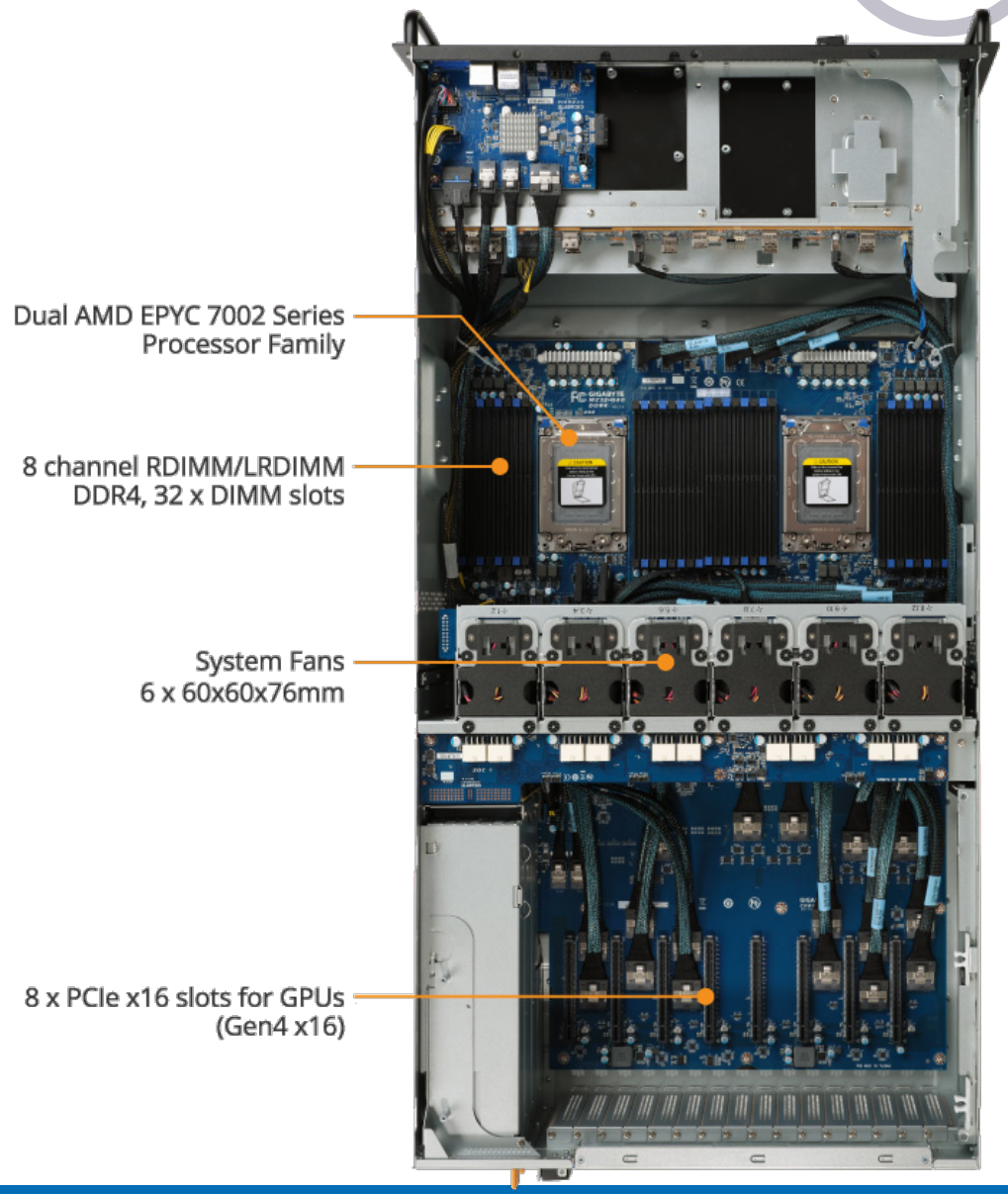
Timing, Synchronous Control & Partitioning

- Synchronously driving the Front-End electronics over the GBT
- 10G-PON for efficient Back-End signal distribution and fixed phase clock recovery
- Partitioning for debugging and commissioning



Event Builder server

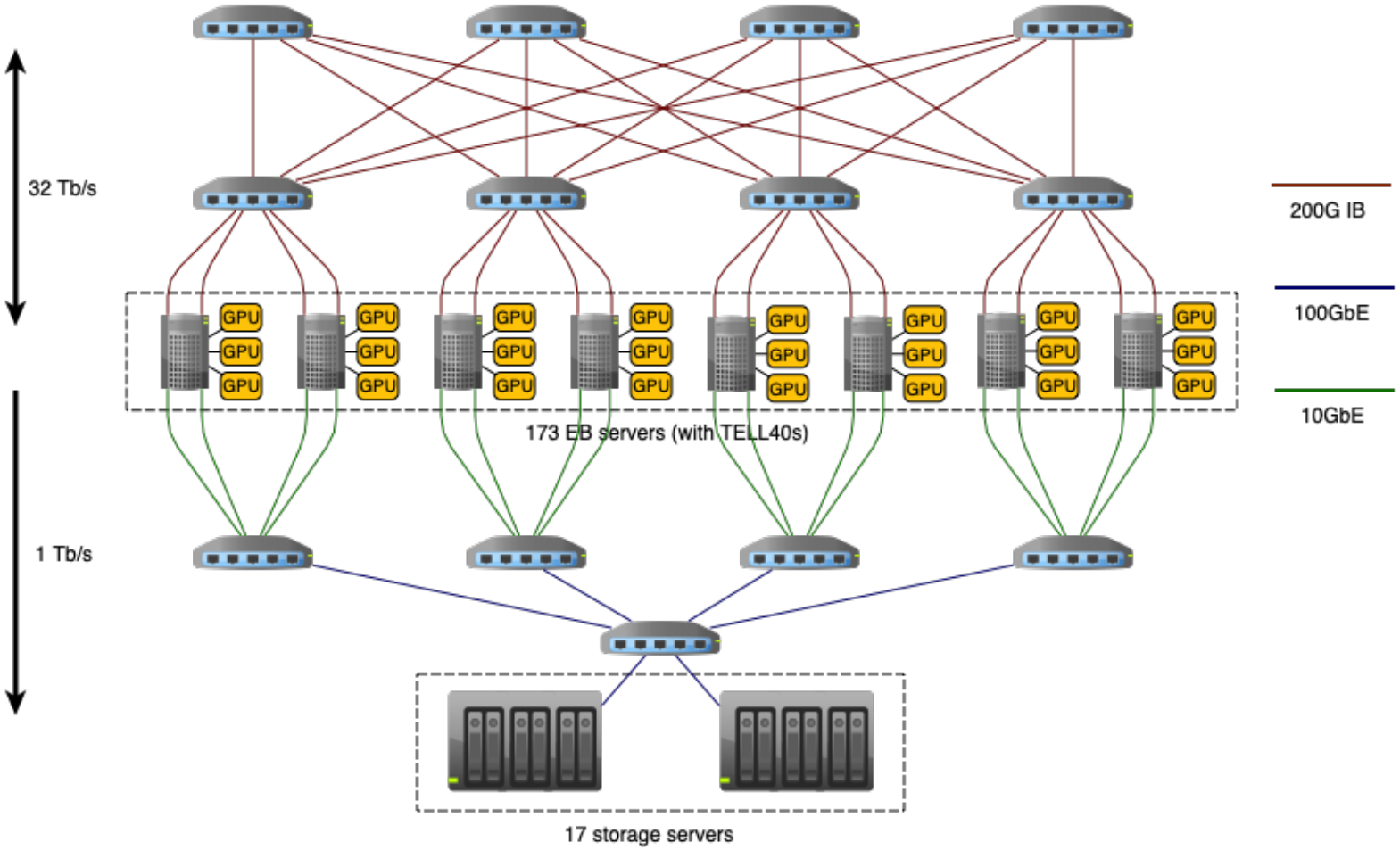
- Dual socket AMD EPYC 7002-series platform:
  - 8 PCIe Gen 4 x16 slots
  - 8+8 DDR4 memory channels
- 3 Readout boards
- 2 InfiniBand HDR NICs (200 Gb/s)
- Up to 3 GPGPUs
- 512 GiB of RAM



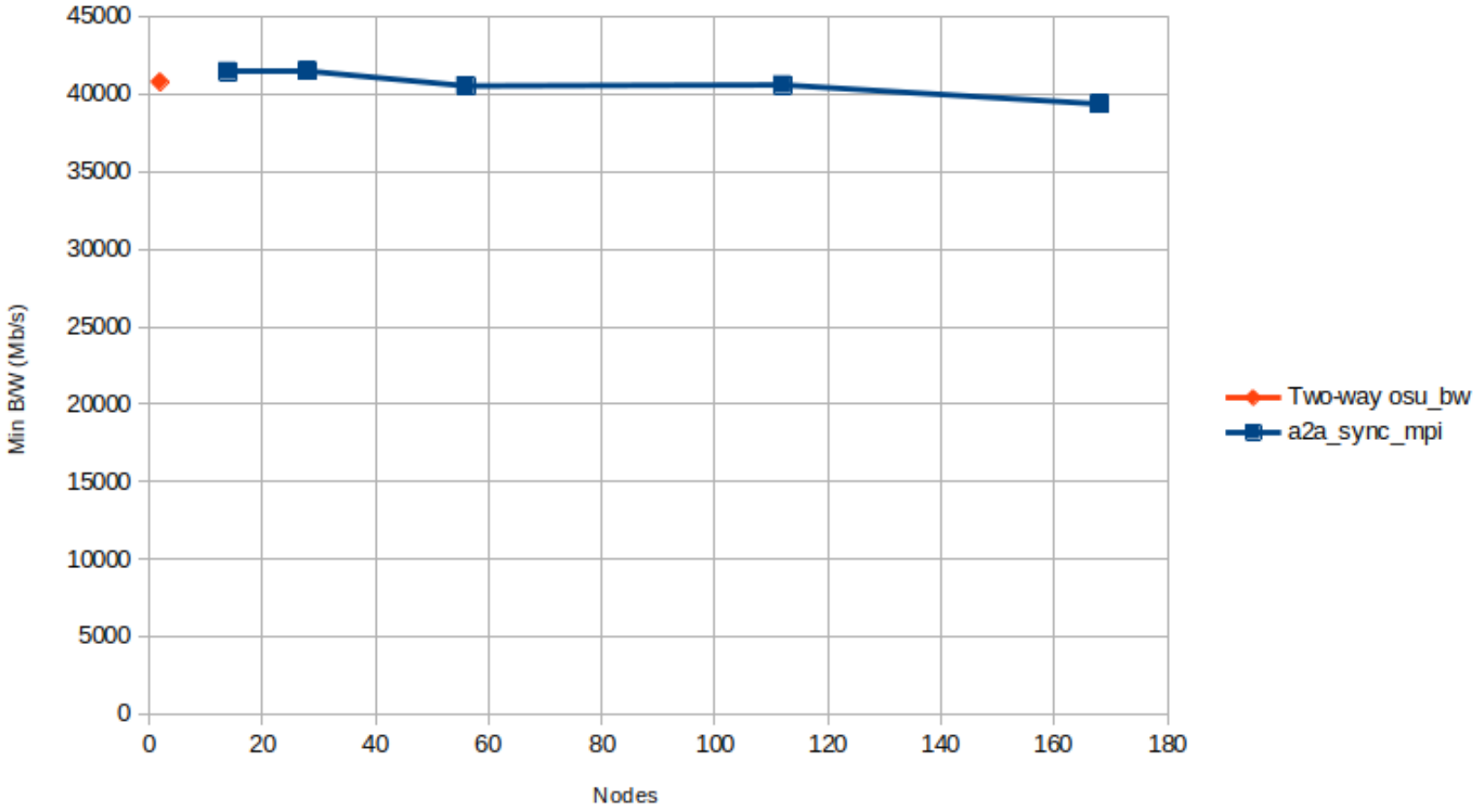
Challenges for the EB network

- It needs to collect data from 478 TELL40 FPGA boards into a single "location"
- And hand it over to GPGPUS + CPUs for further processing
- We want high link-load (more cost effective)
- We want to use some kind of remote DMA to reduce server-load
- Traffic is inherently congestion inducing

Event builder networks



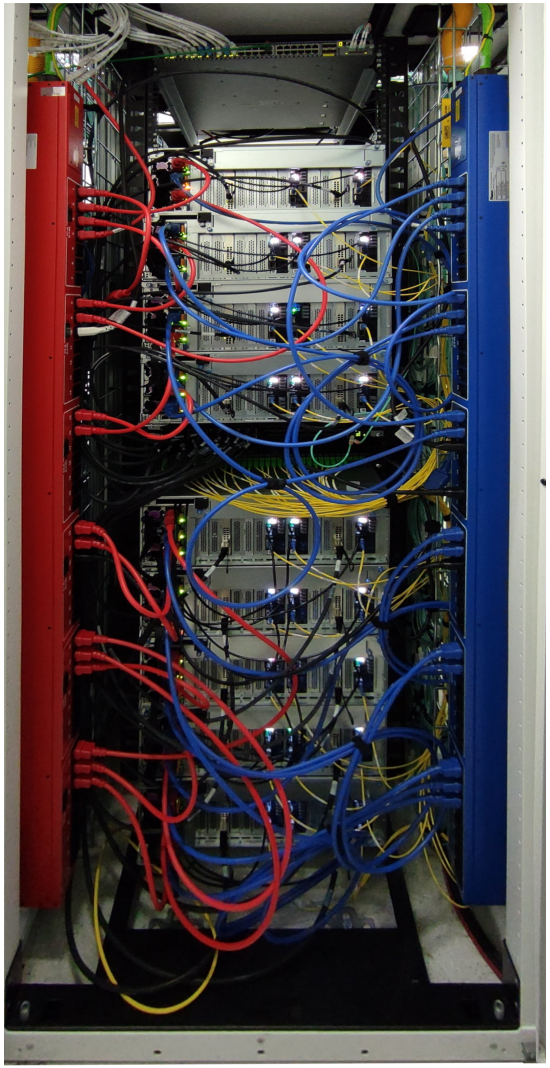
Scalability InfiniBand



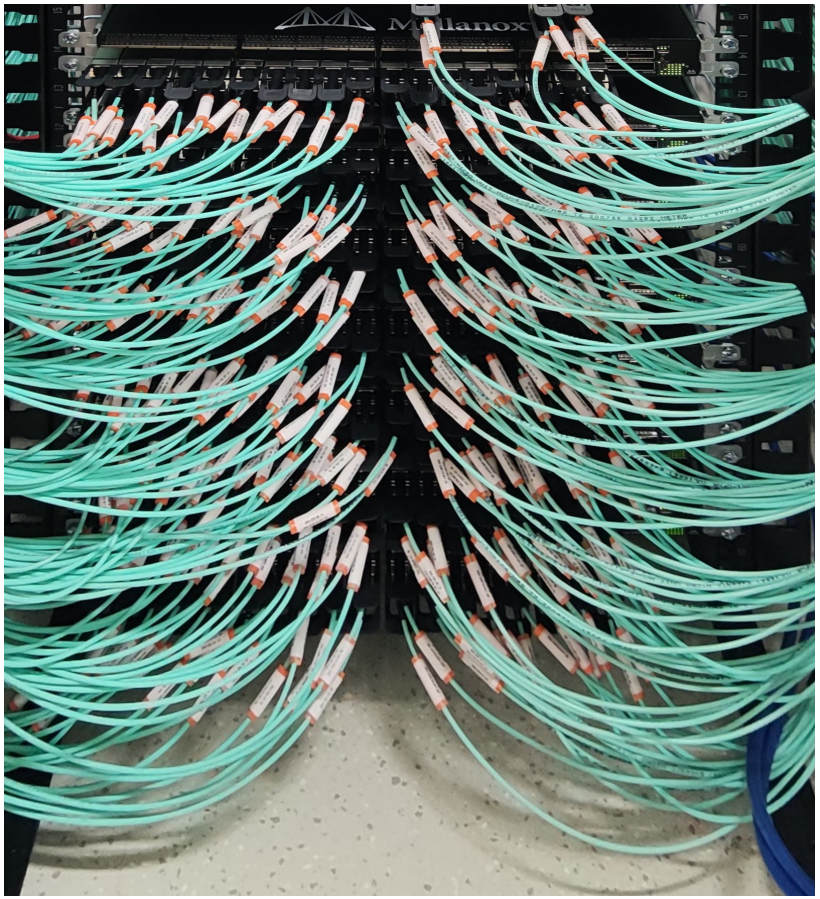
Data collected on the CMS DAQ cluster using InfiniBand FDR (56 Gb/s)

Let's get real

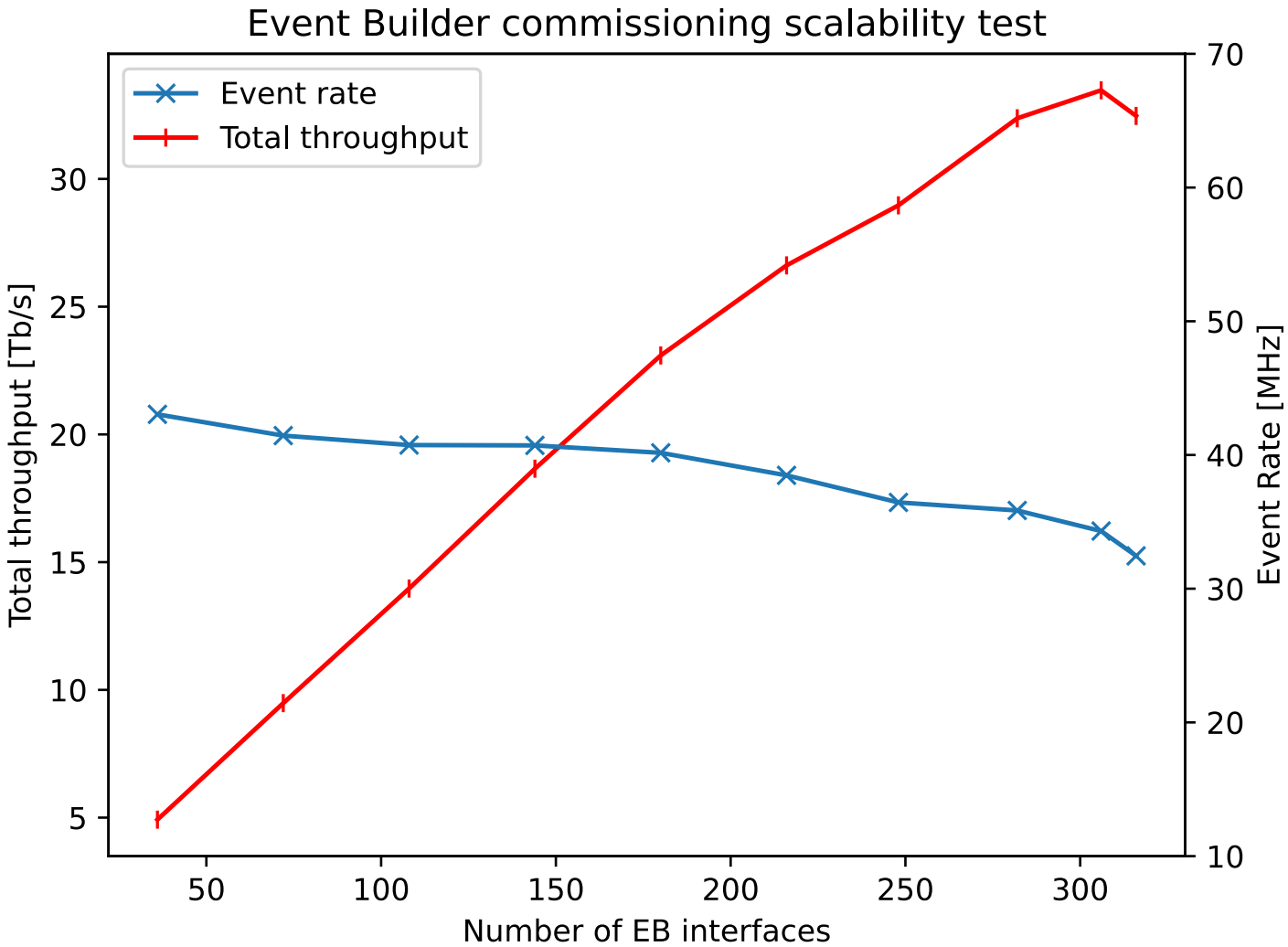
One EB rack



The InfiniBand spine network

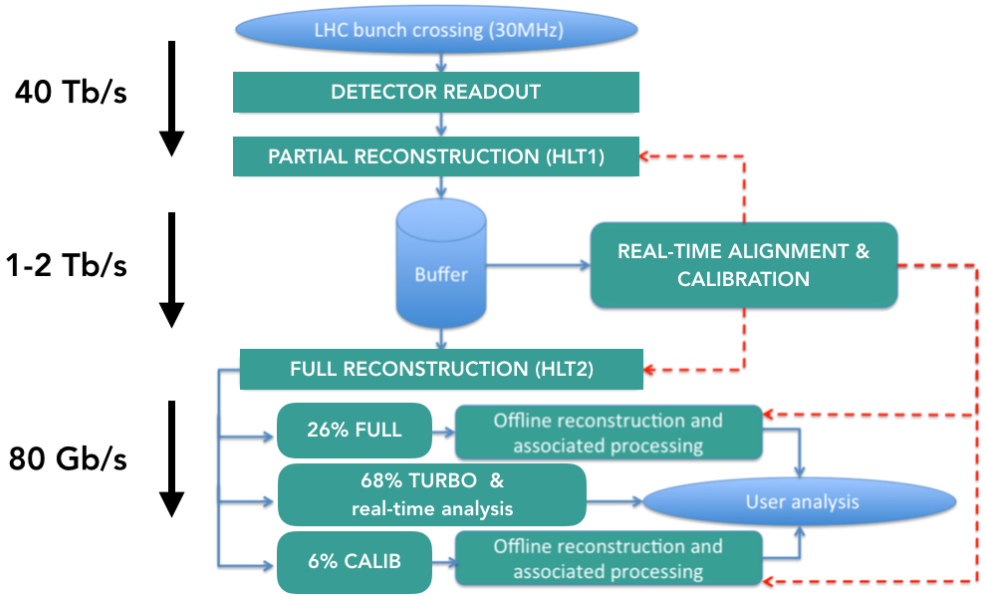


EB commissioning



Data-processing and event selection

- Two stages of software filtering:
  1. "HLT1" on GPGPUs
  2. "HLT2" on a CPU-farm
- Large storage buffer to decouple the two HTL stages
- Calibration and alignment are performed "semi-live", while the data are buffered



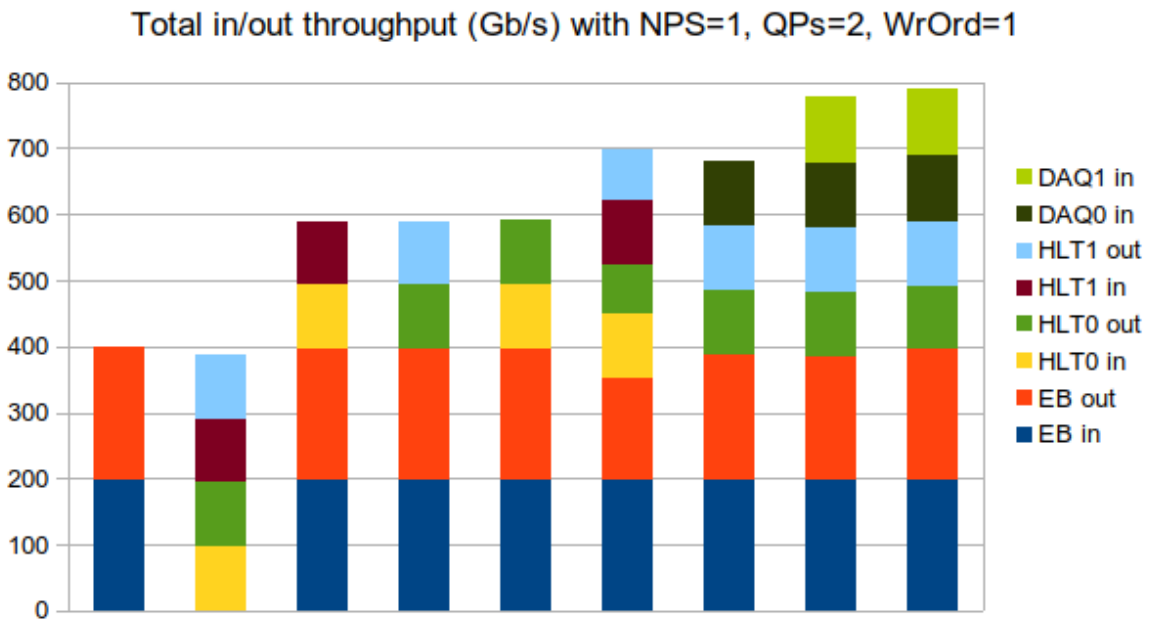
Summary

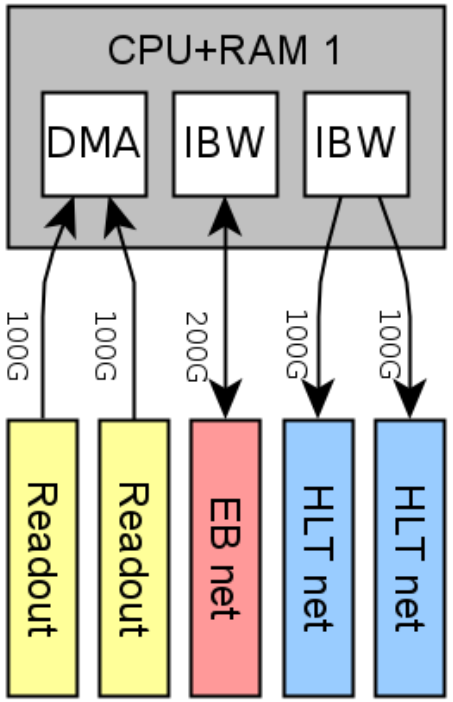
- LHCb can do and afford a full read-out at bunch-crossing rate
- Single stage synchronous readout built around GBT, PON and a single flexible FPGA board
- Detector control uses the same FPGA boards as the timing distribution system
- AMD Rome (PCIe Gen4) based servers make compact, very high-I/O event-builder, connected with 200 Gb/s InfiniBand
- Event-selection is entirely in software to maximise physics yield, increase the amount of data collected, flexibility and minimise cost

THANK YOU FOR YOUR ATTENTION

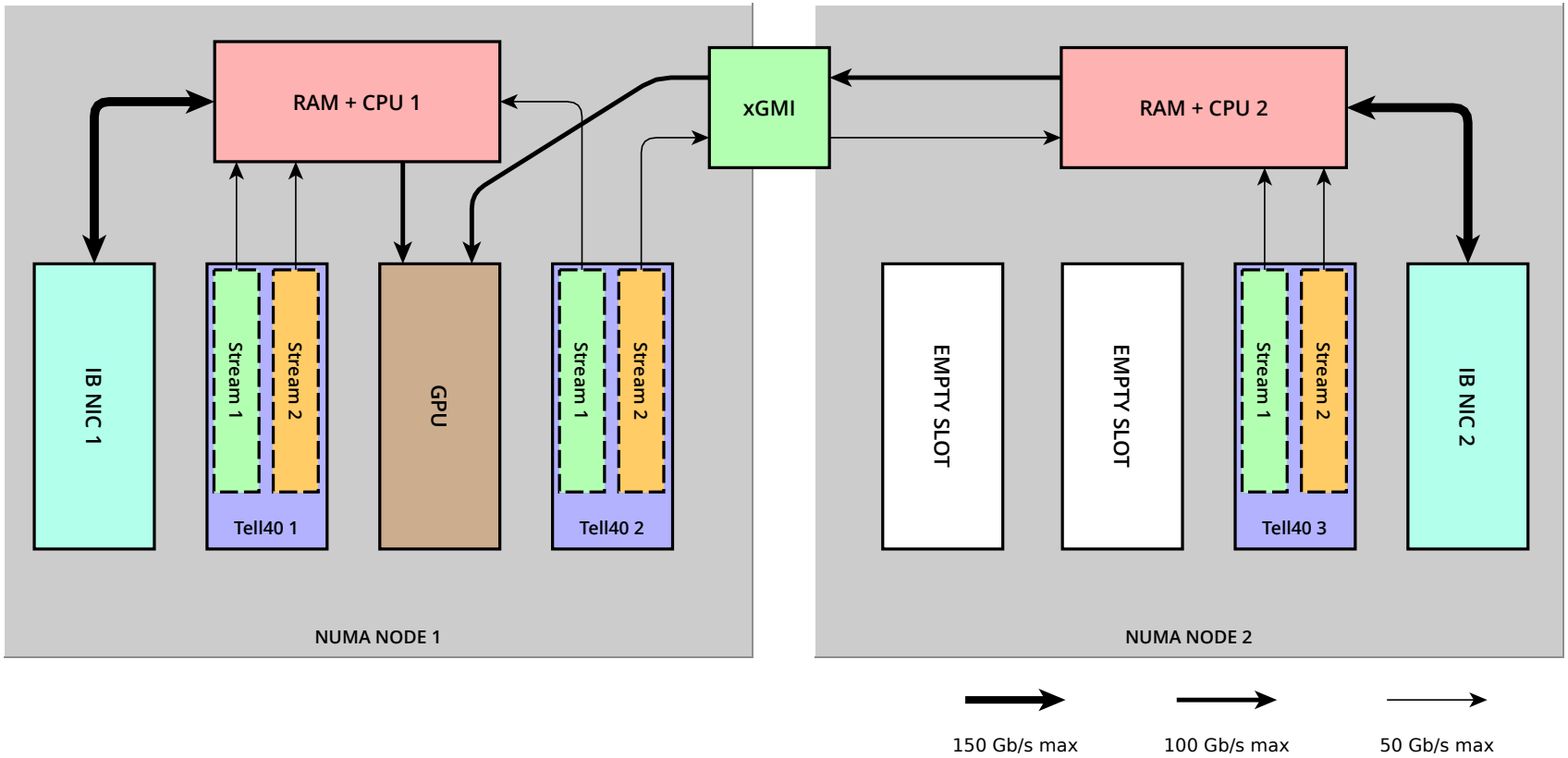
Backup

EB server IO benchmark

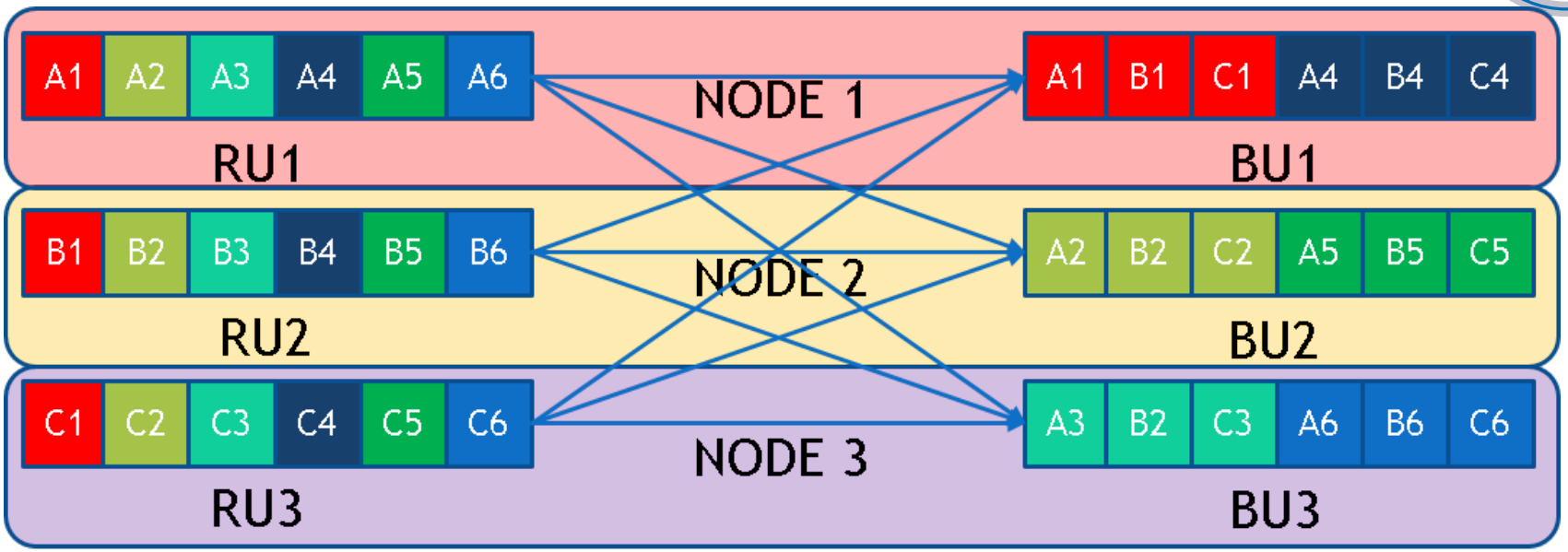




EB server hardware layout



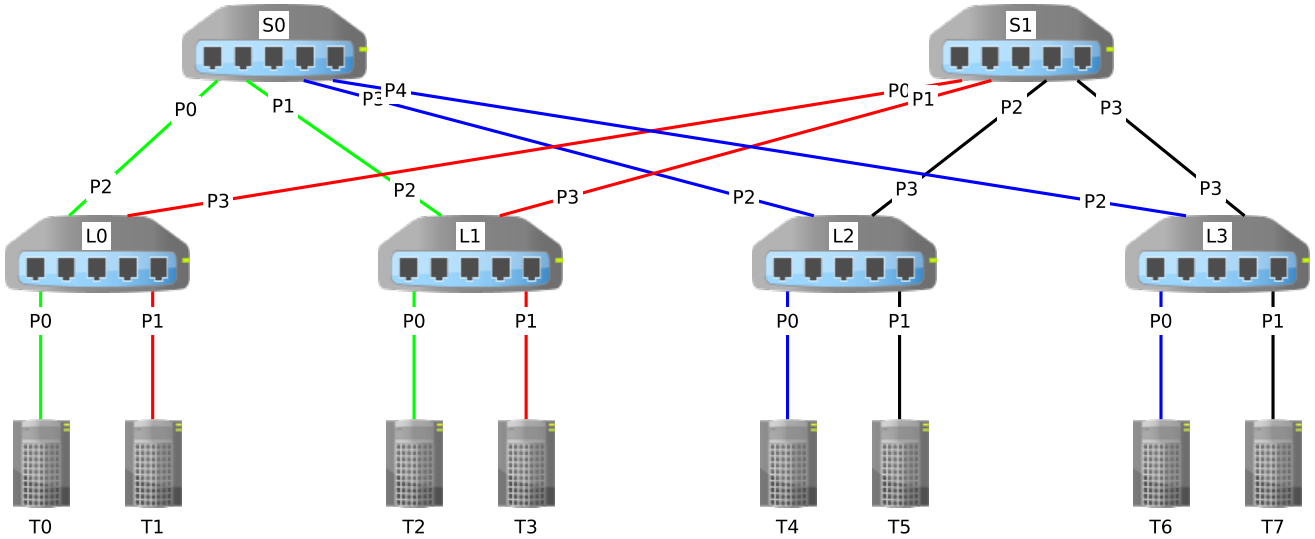
Event Building in a nutshell



- Every event is divided into multiple fragments
- Every **Readout Unit (RU)** receives a fragment of the event
- Every **Builder Unit (BU)** has to gather all the fragments of the event

The many-to-one nature of the traffic generates network congestion

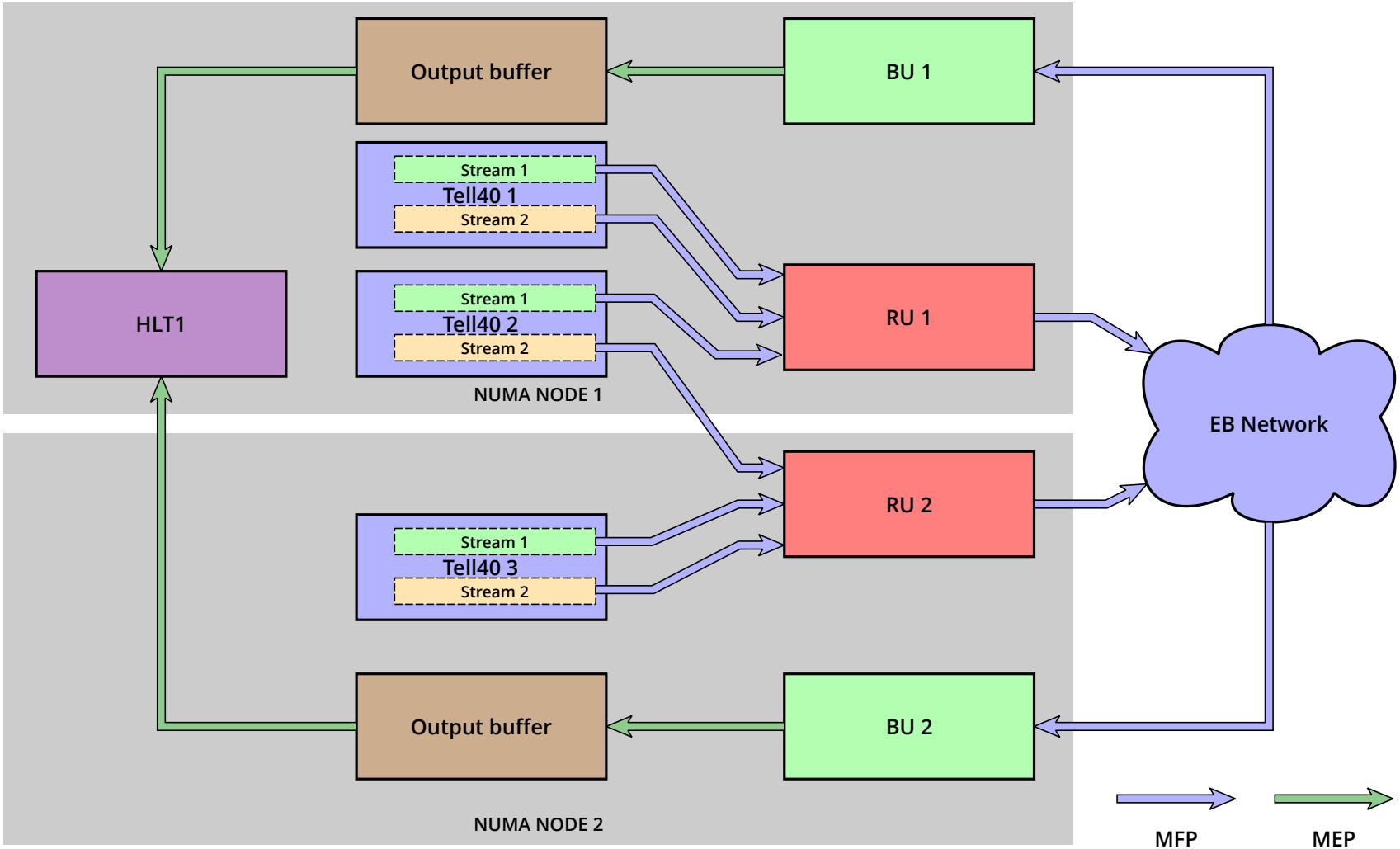
Linear shifting



- The processing of N events is divided into N phases
- In every phase one RU sends data to one BU, and every BU receives data from one RU
- During phase n RU x sends data to BU $(x + n)\%N$
- All the units switch synchronously from phase n to phase $n + 1$

Congestion-free traffic on "selected network" (i.e. non blocking networks)

EB server software data flow



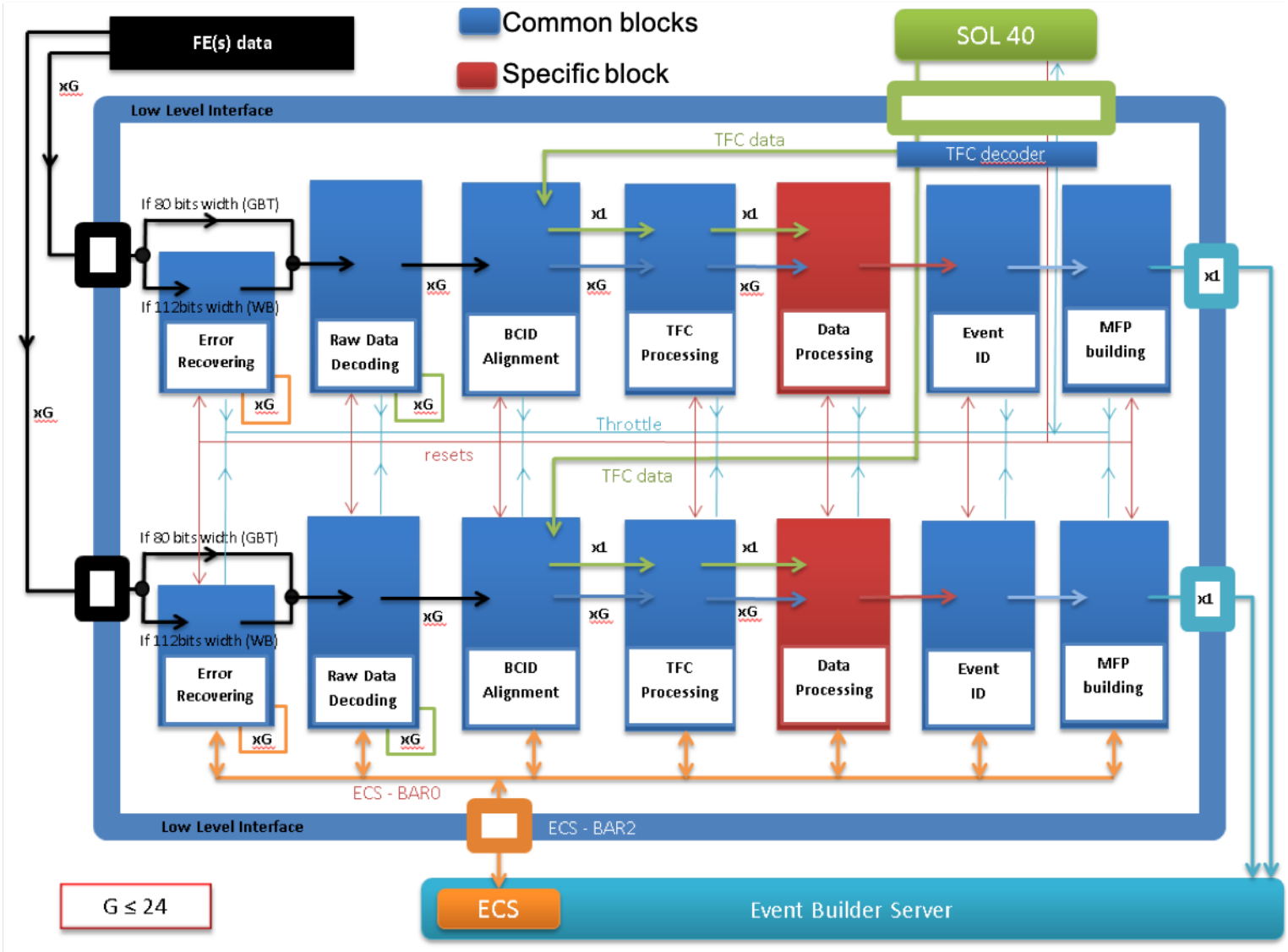
Reasons for going with InfiniBand

- PCIe Gen4 allows using 200 Gbit/s connections which save cost and help with scalability. However 200 Gbit/s so far only effectively exists for InfiniBand!
- Ethernet flow-control could not be made to work properly on available reference platforms
- Ethernet remains - for us - affected by worrying / irritating scaling issues
- Probably most important: could never get access to a really big Ethernet test-system: need the full event-builder for testing. For InfiniBand we have used super-computer sites

...ergo

Lowest risk solution - within our budget - is the InfiniBand solution

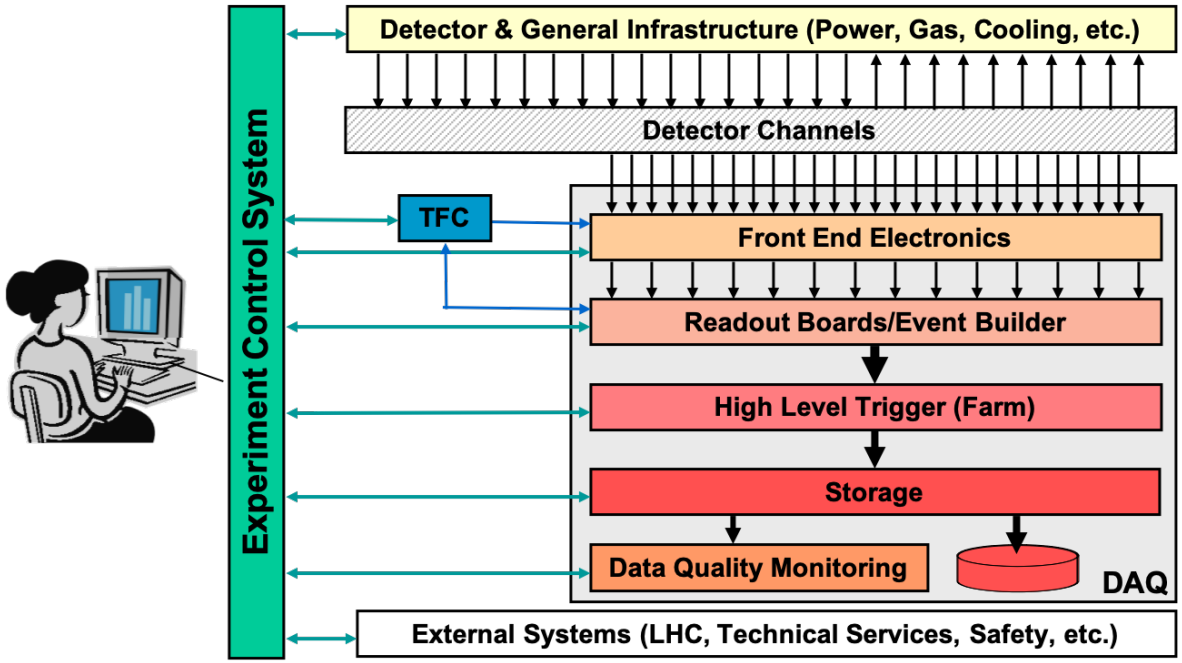
TELL40 architecture



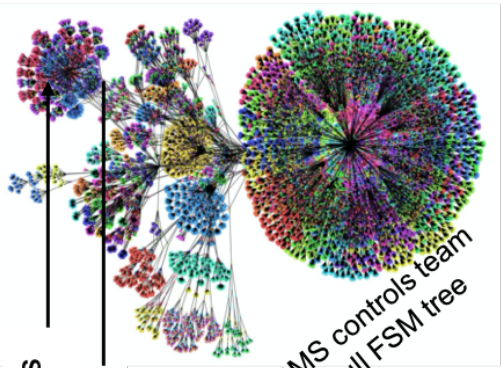
The Experiment Control System

Last but not least

- Operational efficiency of the system is crucial
- LHCb's ECS provides a uniform way to control the **entire** experiment and automate its operation



Uniform, hierarchical control based on FSM



Thanks to the CMS controls team for visualizing our full FSM tree

Status & Alarms

Commands

Scalability Ethernet (deep buffers)