The LHCb trigger and data acquisition system in Run 3

Daniel Hugo Cámpora Pérez

Maastricht University

Department of Data Science and Knowledge Engineering

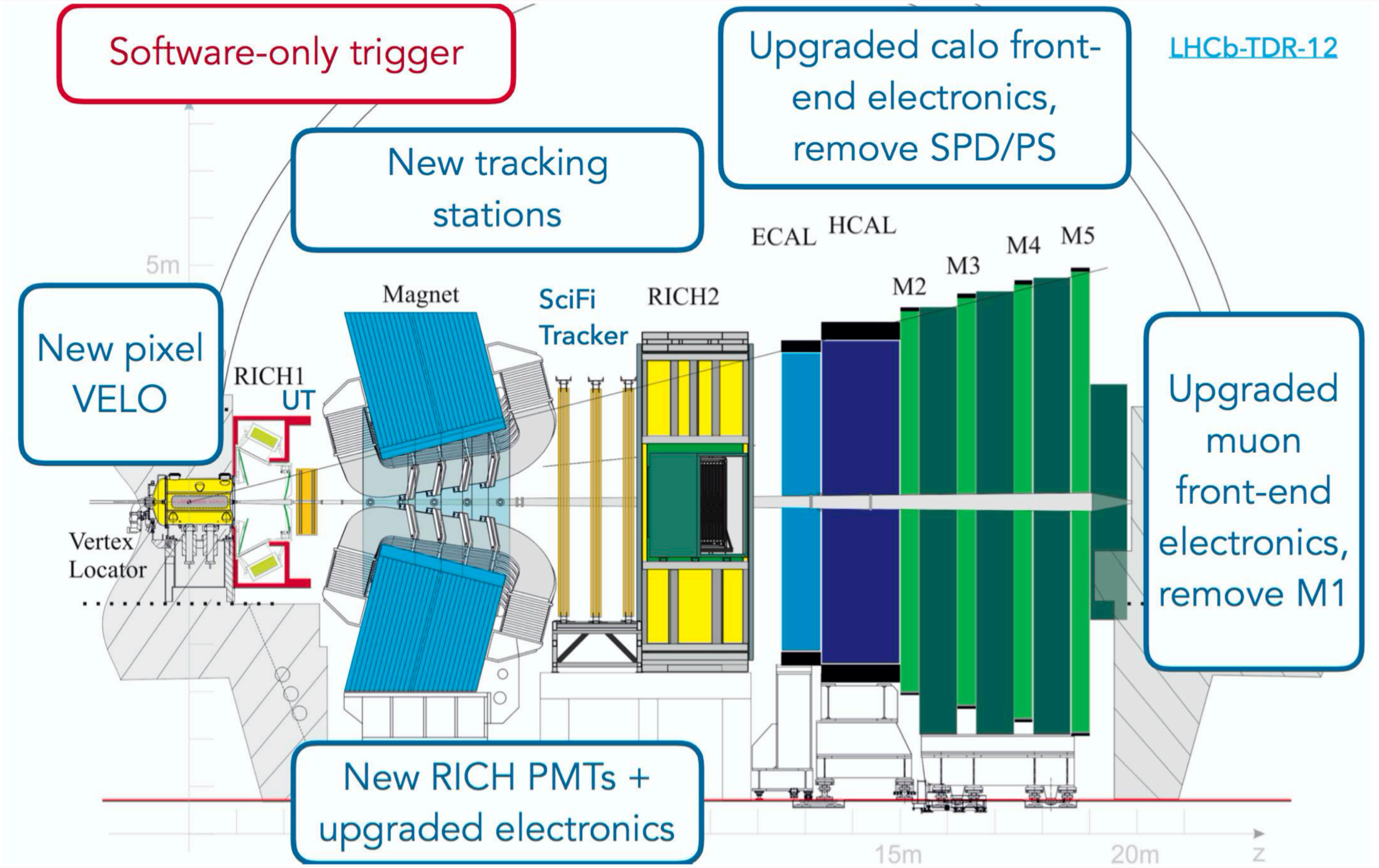
LHCb

About the hardware

About parallel algorithms

About commissioning

The LHCb U1 upgrade



The LHCb detector at CERN:

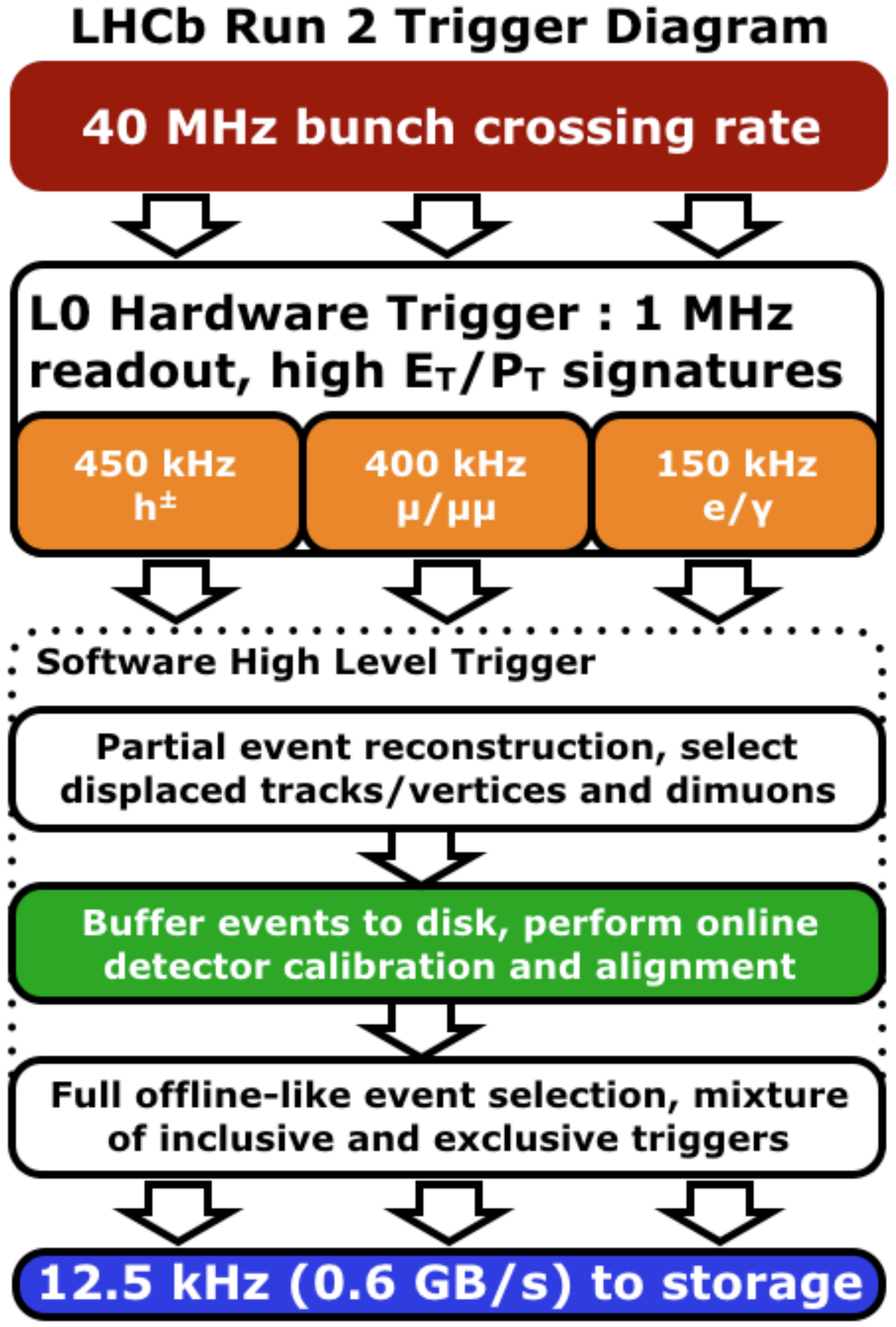
- Single-arm forward spectrometer for high-precision flavour physics
- High precision tracking and vertexing
- Complemented with excellent PID

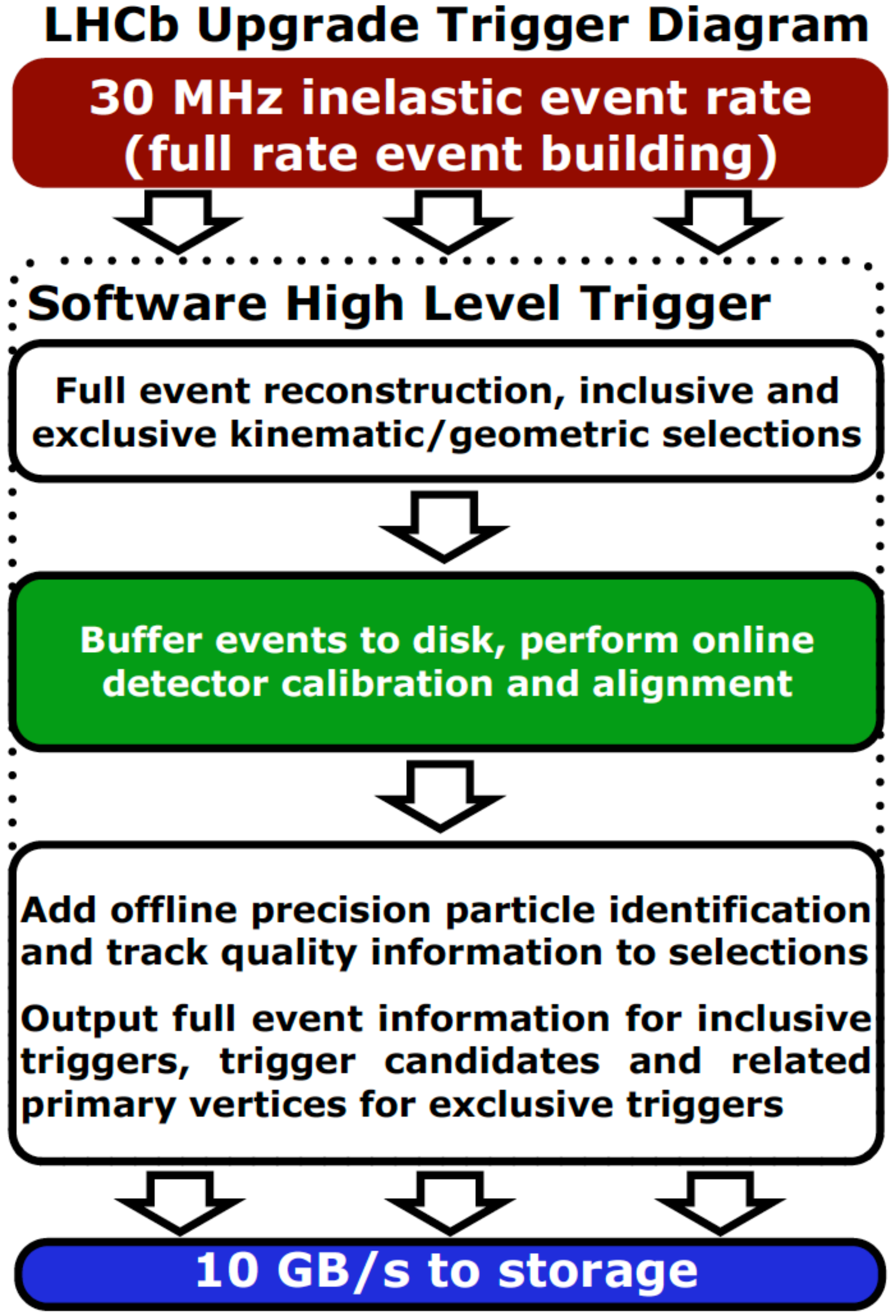
The U1 upgrade

- Instantaneous luminosity will increase by x5
- Major upgrade in all sub-detectors to handle increased rates
- Software-only trigger!

Daniel Cámpora

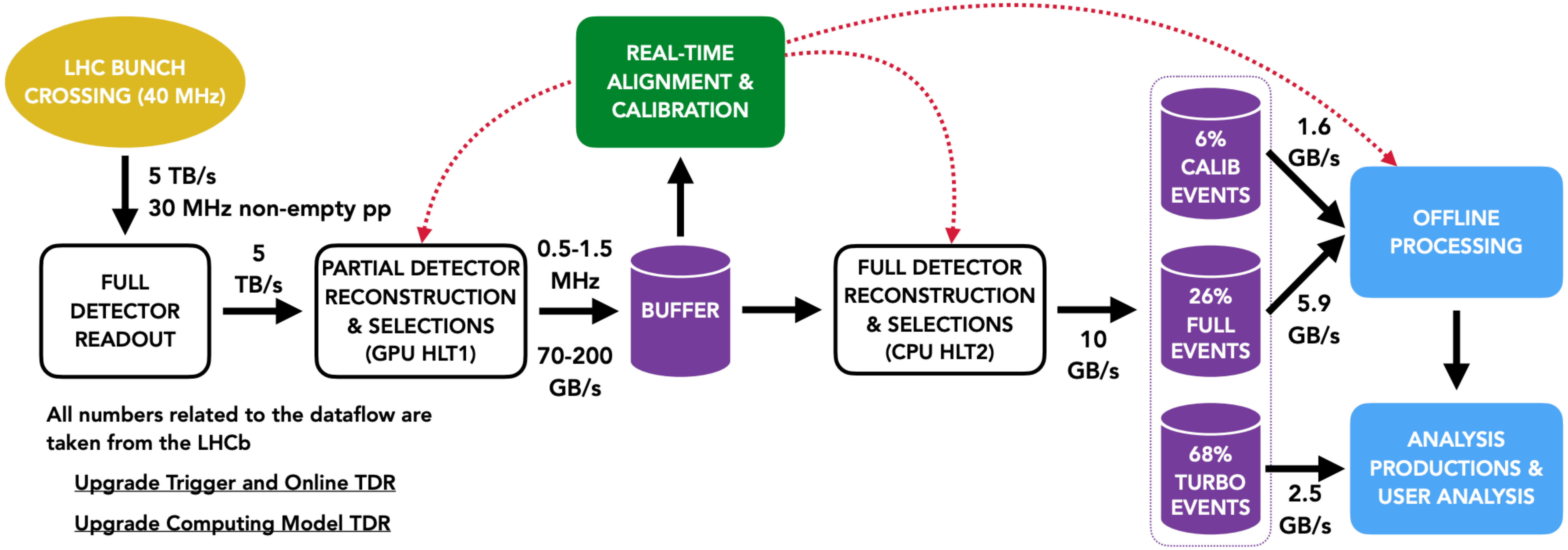
Someone had to pull the trigger





Daniel Cámpora

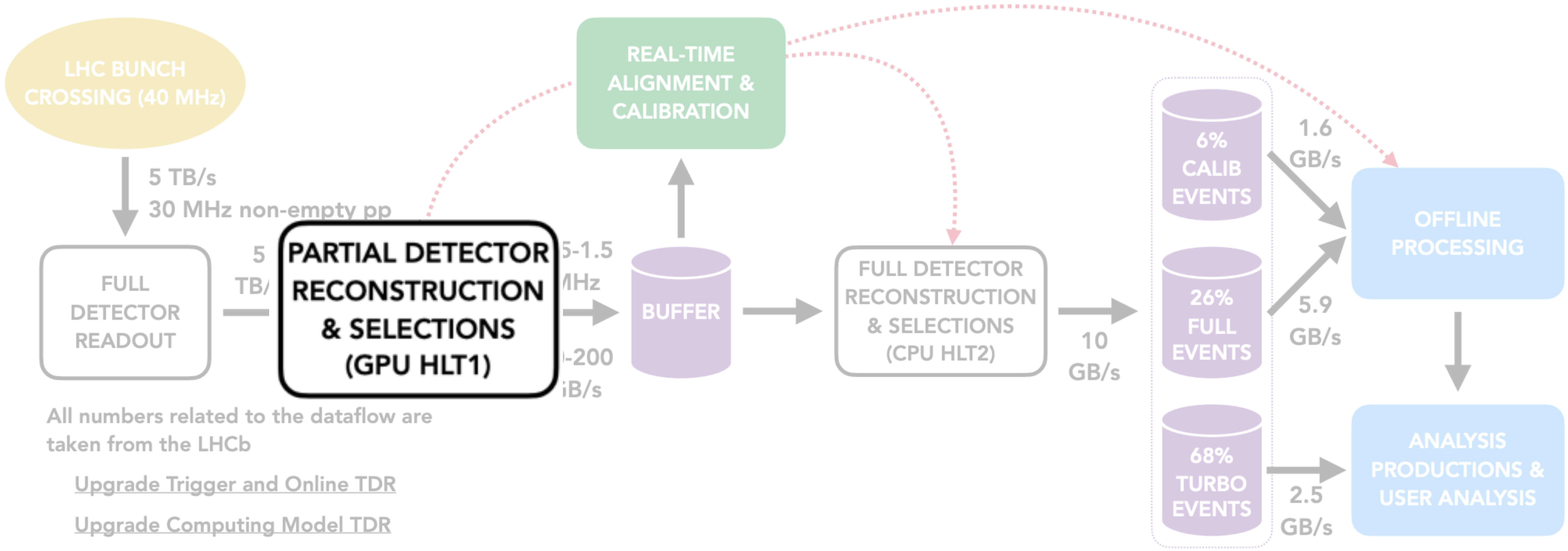
The LHCb data-flow



- Detector data received by O(500) FPGAs and built into events in the event building (EB) farm servers
- 2-stage software trigger, HLT1 & HLT2
- Real-time alignment & calibration
- After HLT2, 10 GB/s of data for offline processing

Daniel Cámpora

The LHCb first level trigger

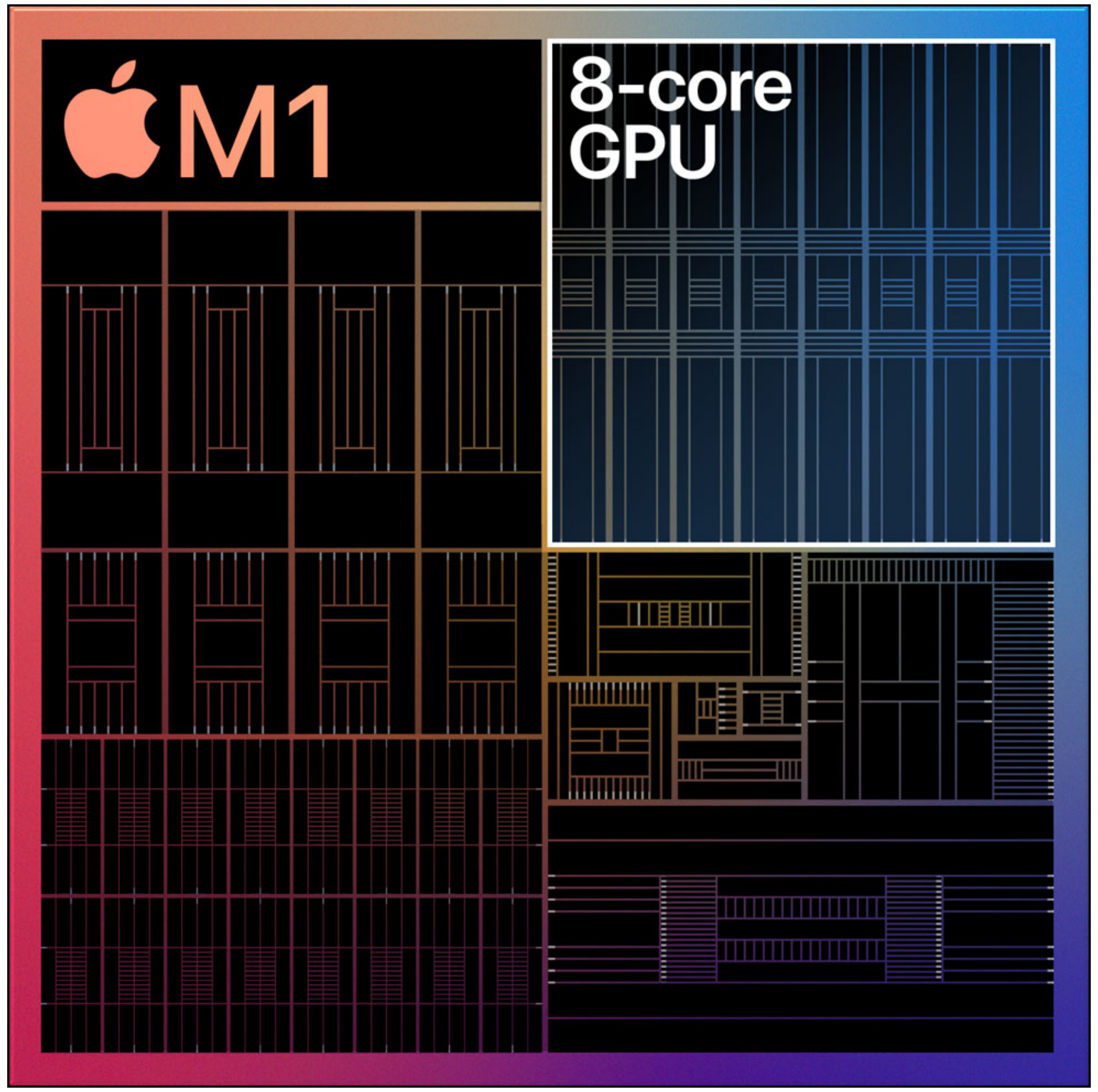


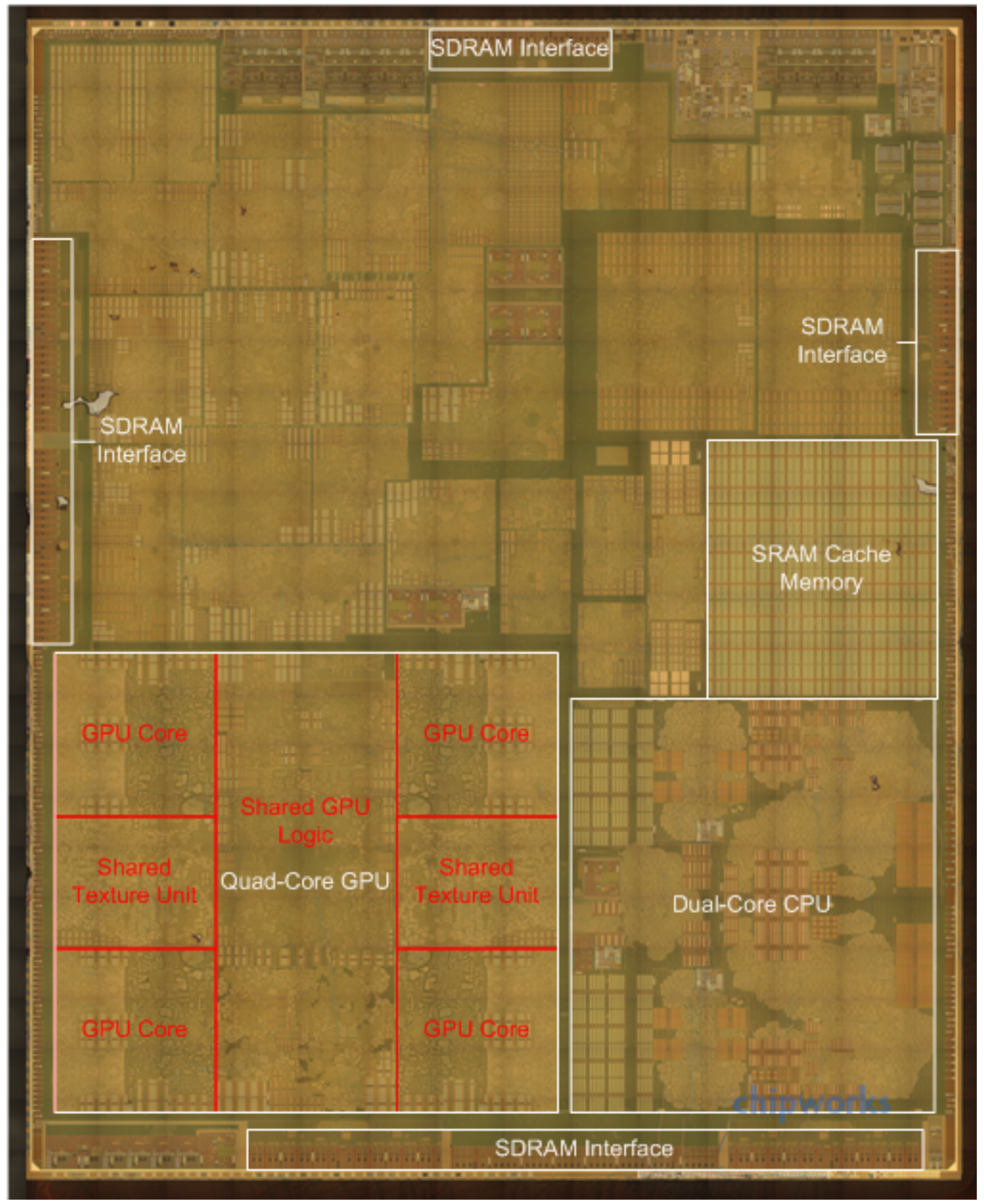
- **The goal of HLT1:**
  - Be able to intake the entirety of the LHCb raw data (5 TB/s) at 30 MHz
  - Perform partial event reconstruction & coarse selection of broad LHCb physics cases
  - Reduce the input rate by a factor of 30 (~ 1 MHz)
  - Store selected events in intermediate buffer for real-time alignment and calibration

But why GPUs?

Daniel Cámpora

CPU radiography





Daniel Cámpora

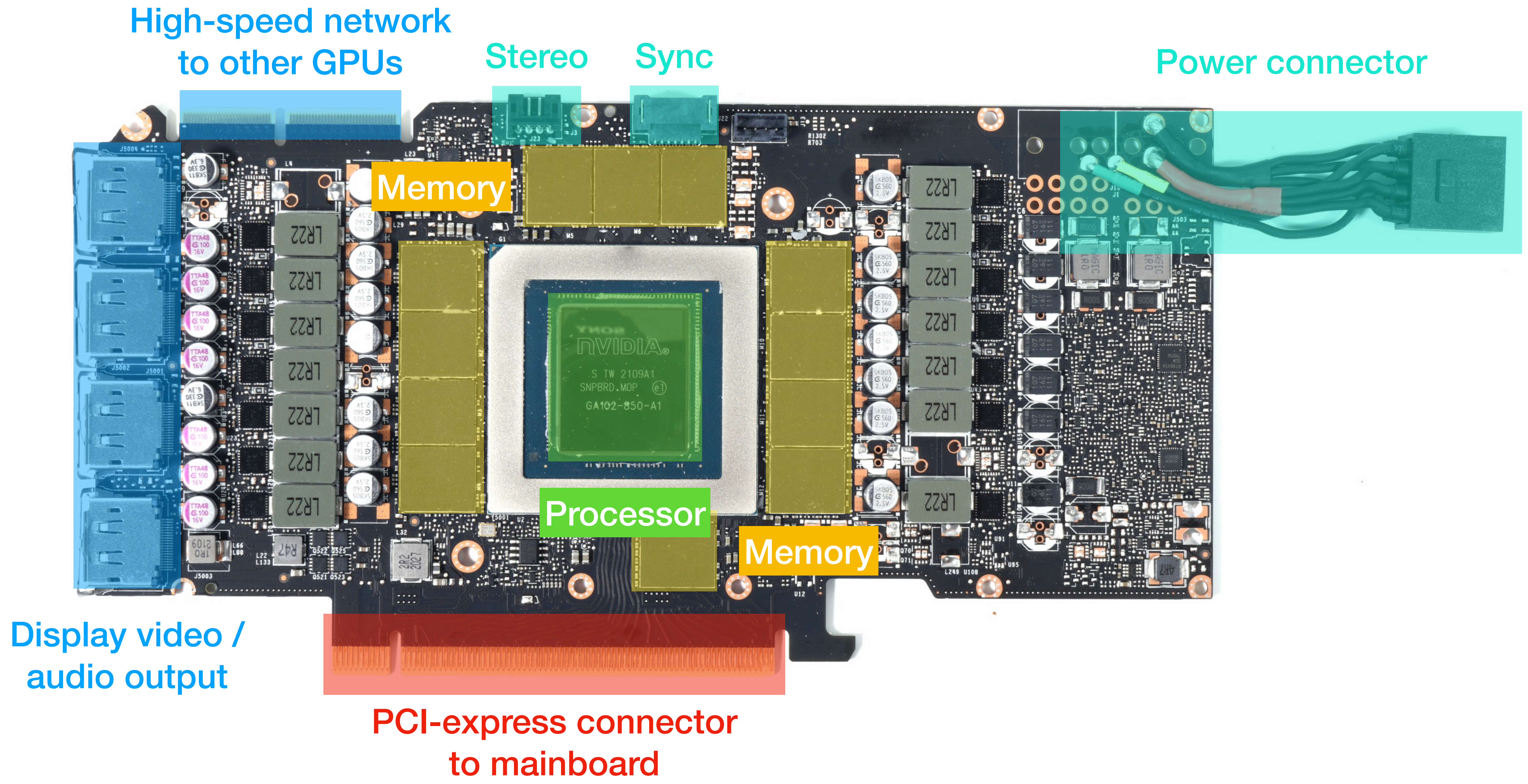
GPUs: Parallel processors

- GPUs are processors specialized to perform graphic-oriented workloads

Daniel Cámpora

How does a dedicated GPU card look like?

How does a dedicated GPU card look like?



Daniel Cámpora

Strengths and limitations

Memory	
Memory Size:	24 GB
Memory Type:	GDDR6
Memory Bus:	384 bit
Bandwidth:	768.0 GB/s

Render Config	
Shading Units:	8192
TMUs:	256
ROPs:	96
SM Count:	64
Tensor Cores:	256
RT Cores:	64
L1 Cache:	128 KB (per SM)
L2 Cache:	6 MB

Theoretical Performance	
Pixel Rate:	162.7 GPixel/s
Texture Rate:	433.9 GTexel/s
FP16 (half) performance:	27.77 TFLOPS (1:1)
FP32 (float) performance:	27.77 TFLOPS
FP64 (double) performance:	867.8 GFLOPS (1:32)

https://www.techpowerup.com/gpu-specs/rtx-a5000.c3748

- Excellent FP16 or FP32 performance. Avoid FP64.
- Room for growth: Tensor cores, RT cores.
- Limited memory: 24000 / 8192 = 2.9 MB per core.

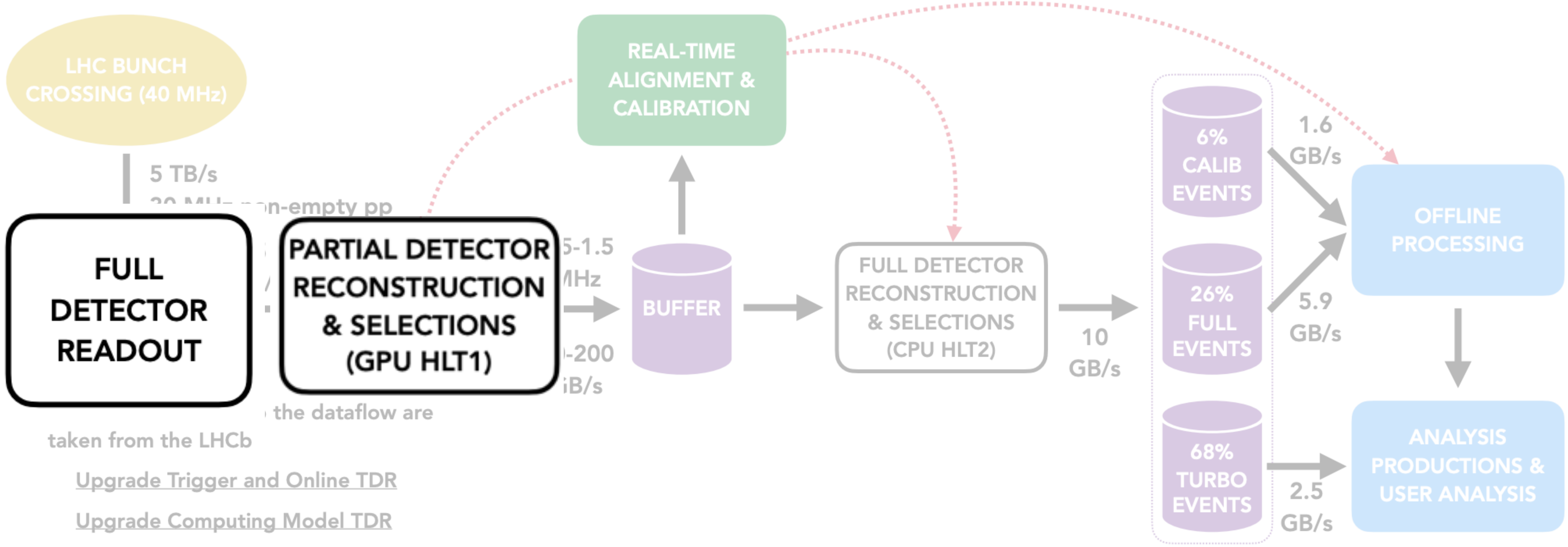
Daniel Cámpora

The main concepts behind the GPU HLT1

- Pipeline: raw-data in, selections out
- Scheduler: many events run concurrently
- Redesign algorithms:
  - Memory: very limiting O(MB) per core
  - Parallelism

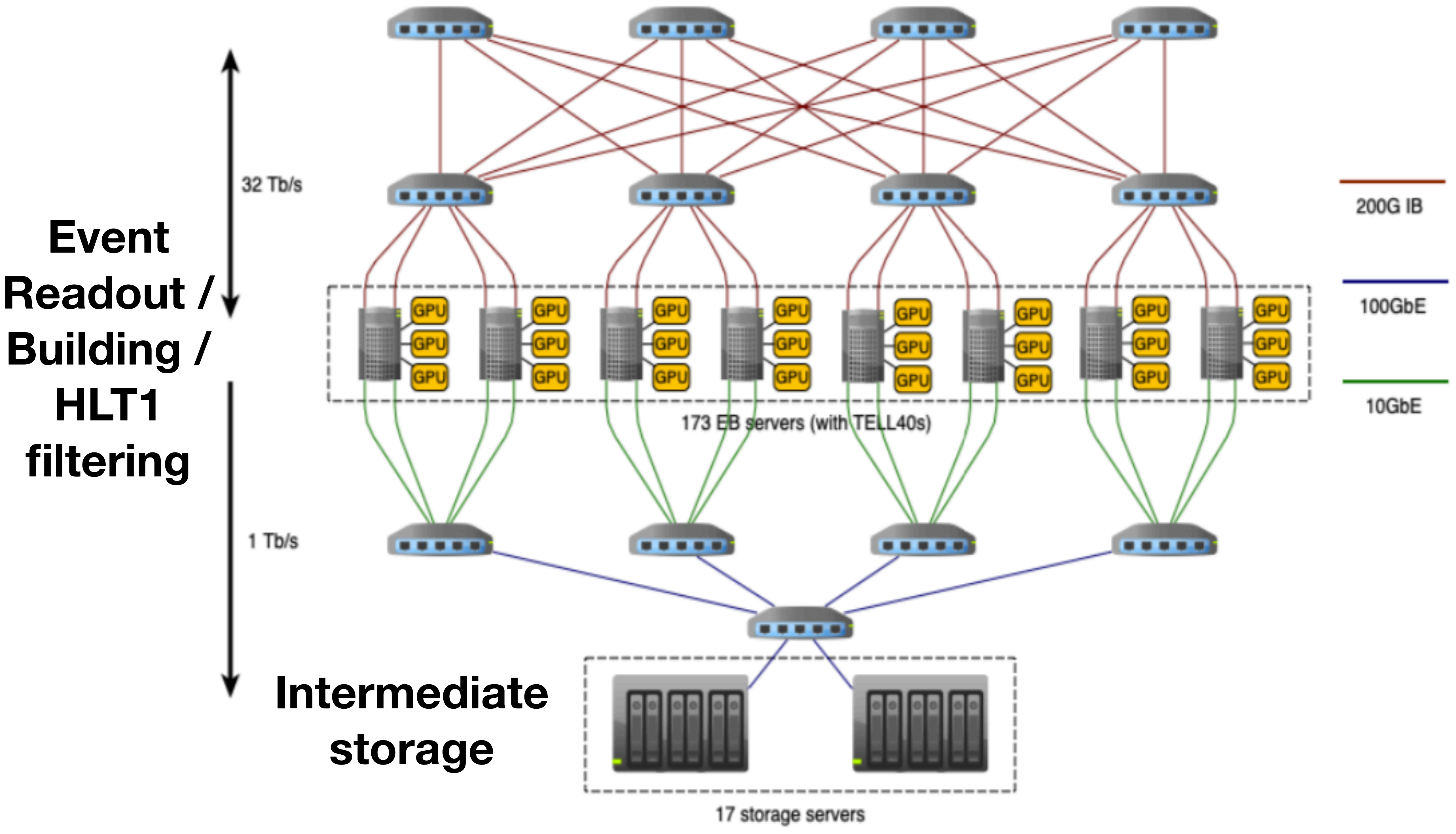
Daniel Cámpora

The LHCb DAQ



Daniel Cámpora

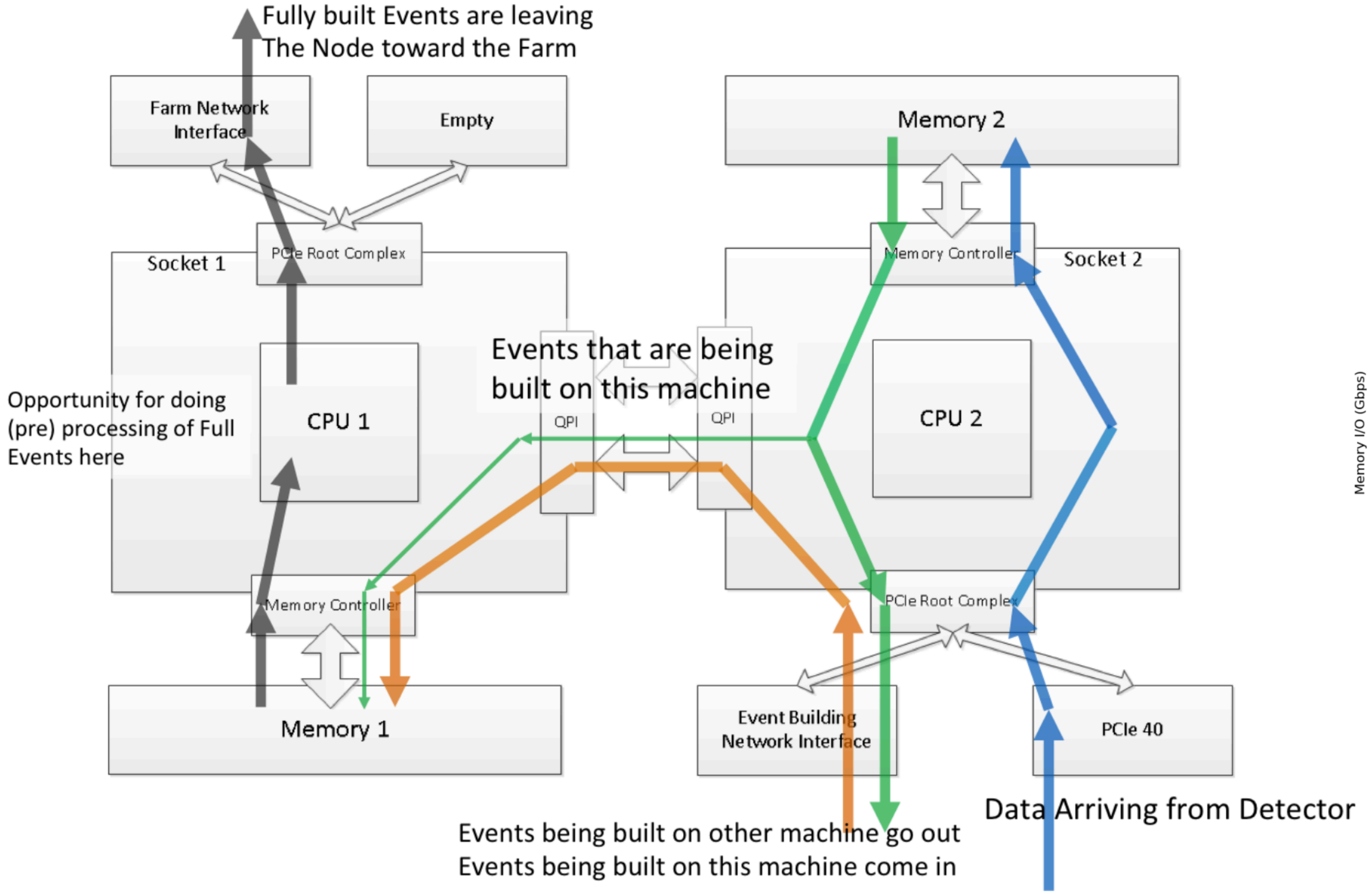
Event Building

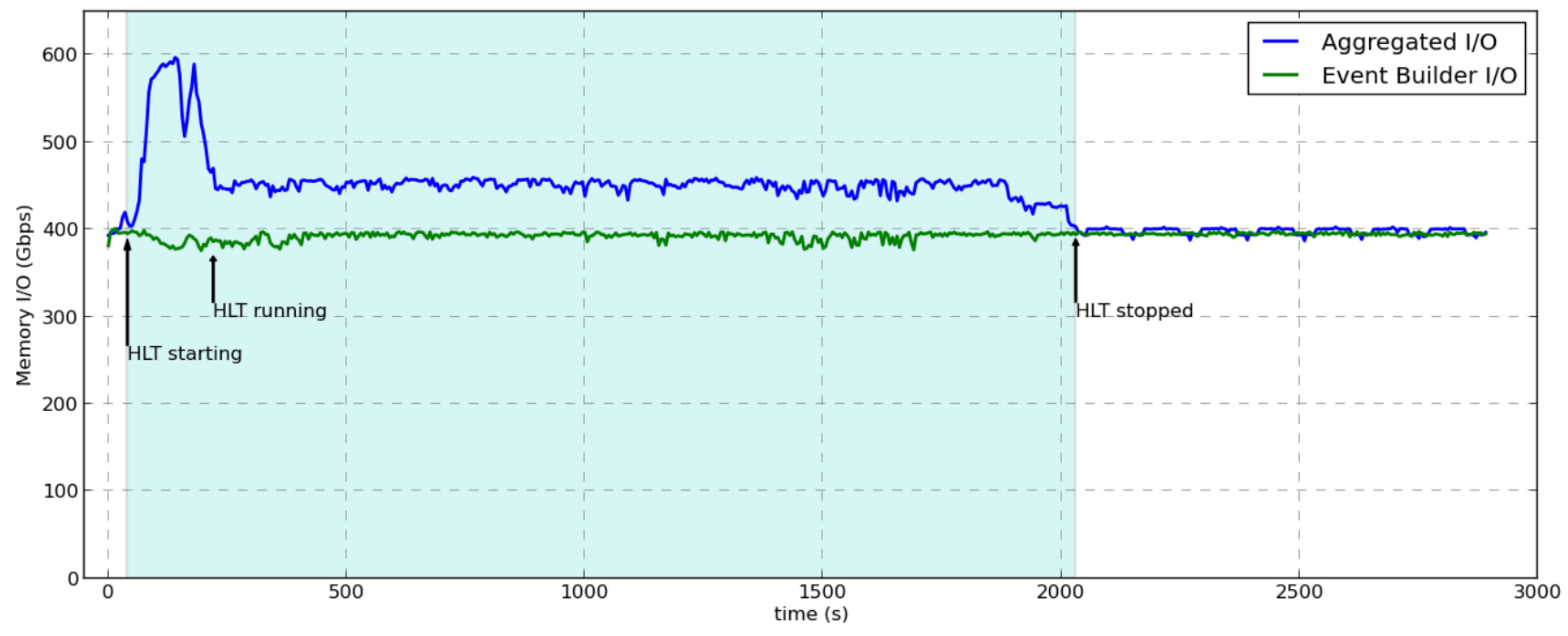


- Event fragments are collected from the detector in readout cards (PCIe40)
- Data from these fragments is distributed to one destination at a time
- Events are fully built
- Events are processed by HLT1
- Finally, events are sent to the Event Filter Farm

Daniel Cámpora

It works! (2014)





http://cds.cern.ch/record/1701361

Putting all together

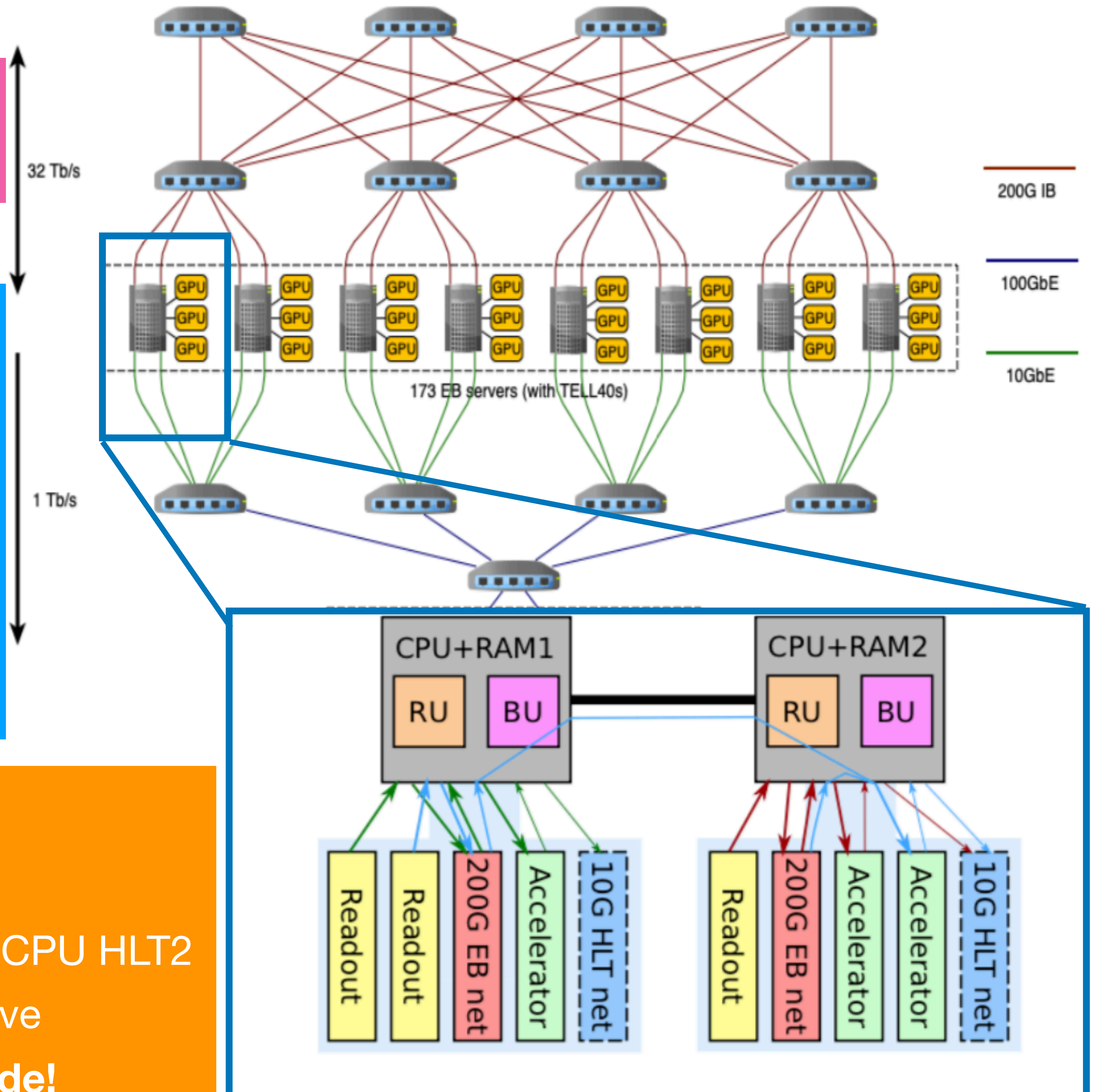
- Event builder farm equipped with 173 servers

- Each server has 3 free PCIe slots
  - Can be used to host GPUs
  - Sufficient cooling & power
  - Advantageous to have GPUs as self-contained processors
  - Sending data to GPU is like sending data to network card

- GPUs map well into LHCb DAQ architecture
- HLT1 tasks inherently parallelizable
- Reduced bandwidth network between EB & CPU HLT2
- Cheaper & more scalable than CPU alternative

➡**Was chosen as the baseline for the upgrade!**

➡**Is implemented with O(200) Nvidia RTX A5000 GPUs**



GPU-equipped event builder PC, with traffic of all three readout cards.

Daniel Cámpora

About the hardware

About parallel algorithms

About commissioning

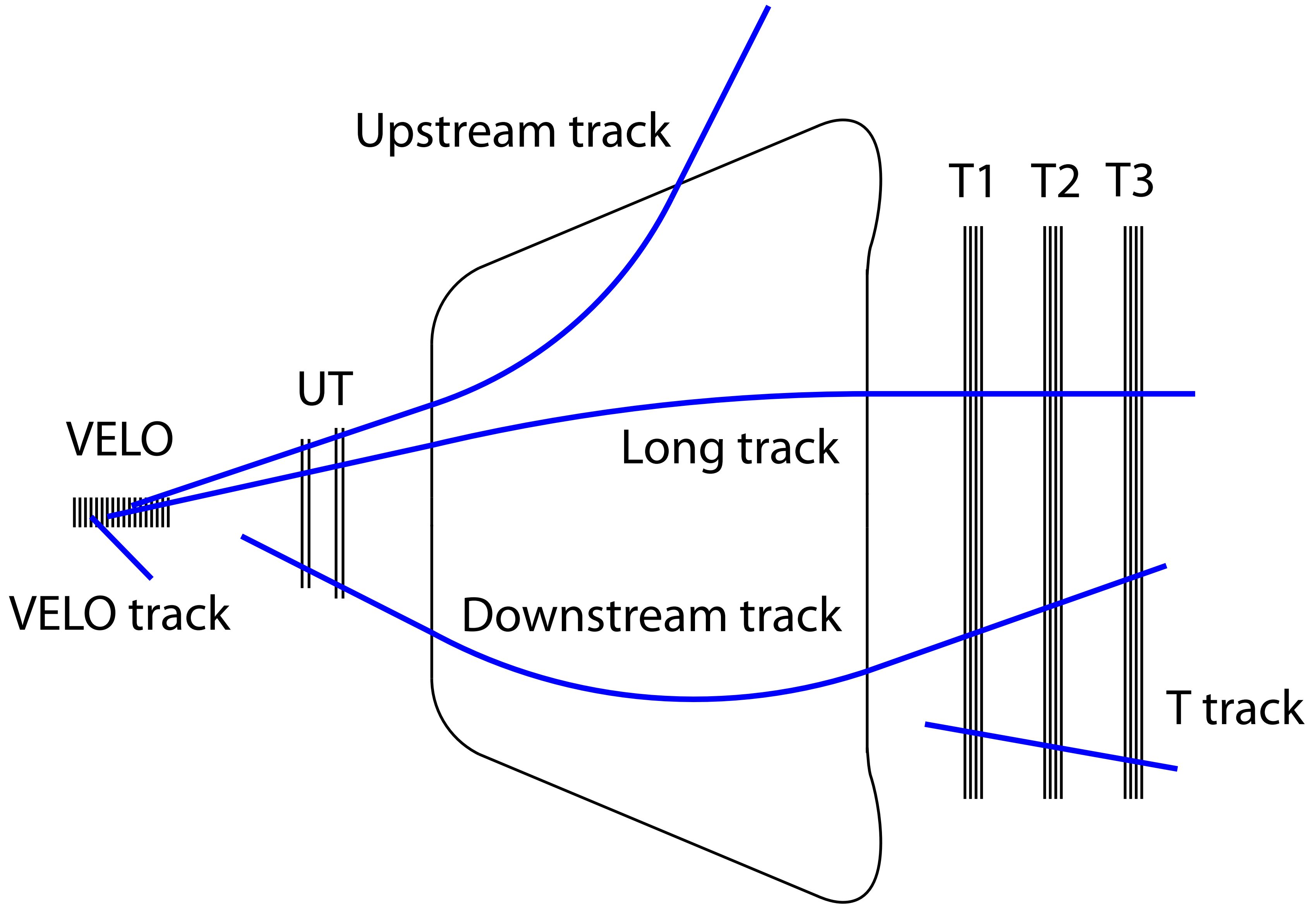
How to make a good parallel algorithm

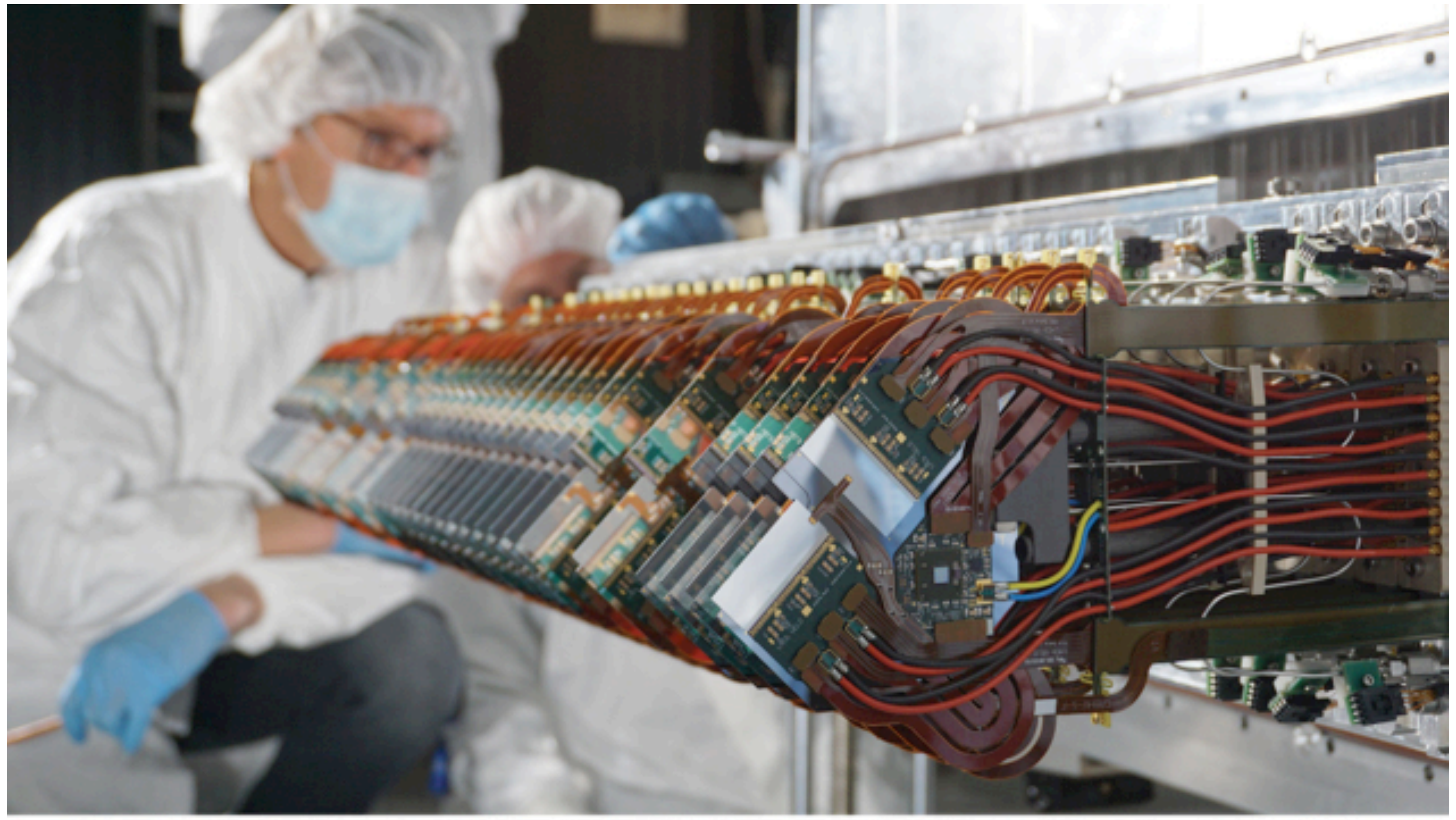
- What degrees of parallelism does your problem have?
- How could you map this parallelism onto your hardware architecture?

- What memory patterns can you identify?
- How can you map these patterns onto your hardware architecture?

Daniel Cámpora

Track finding at LHCb

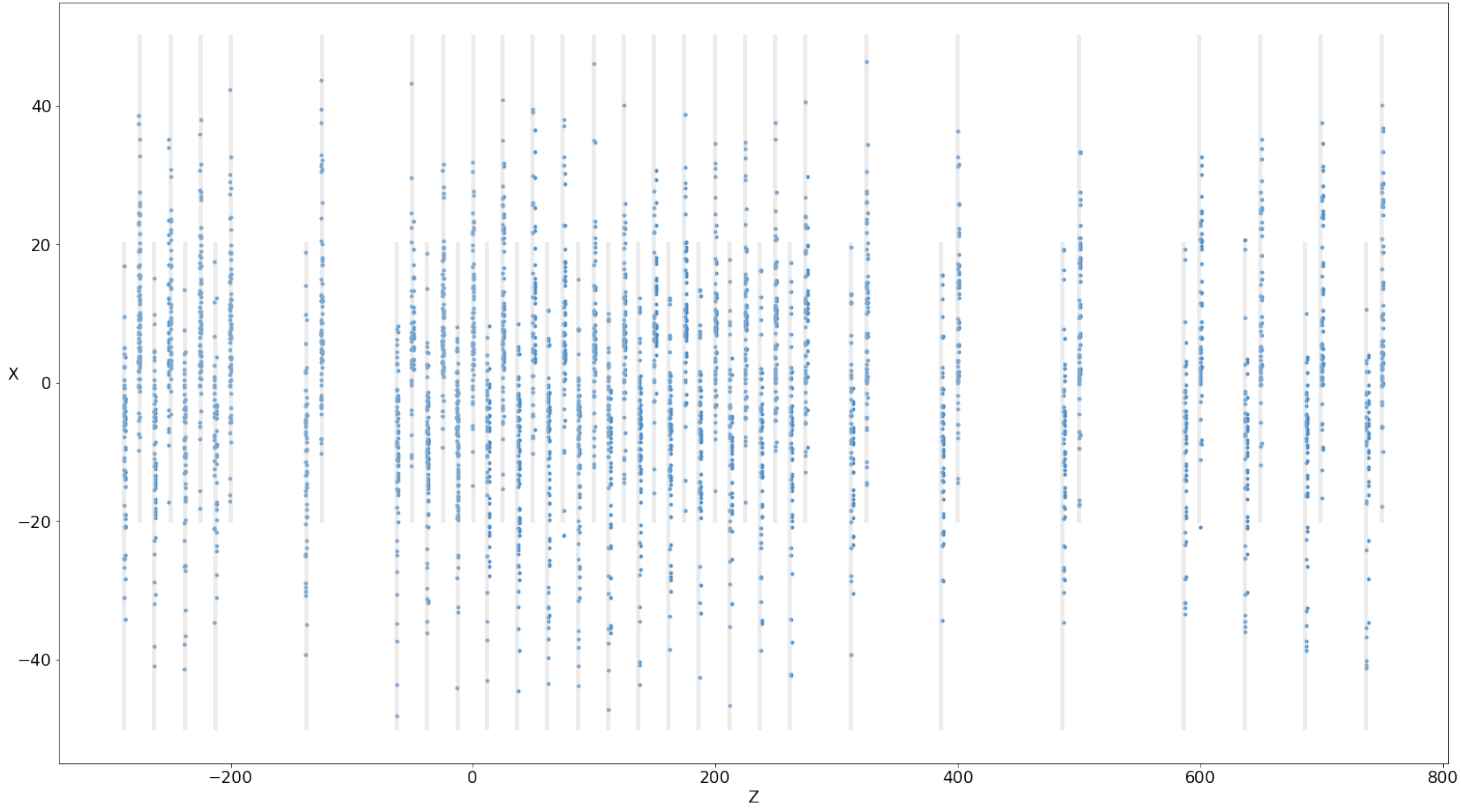




VELO subdetector

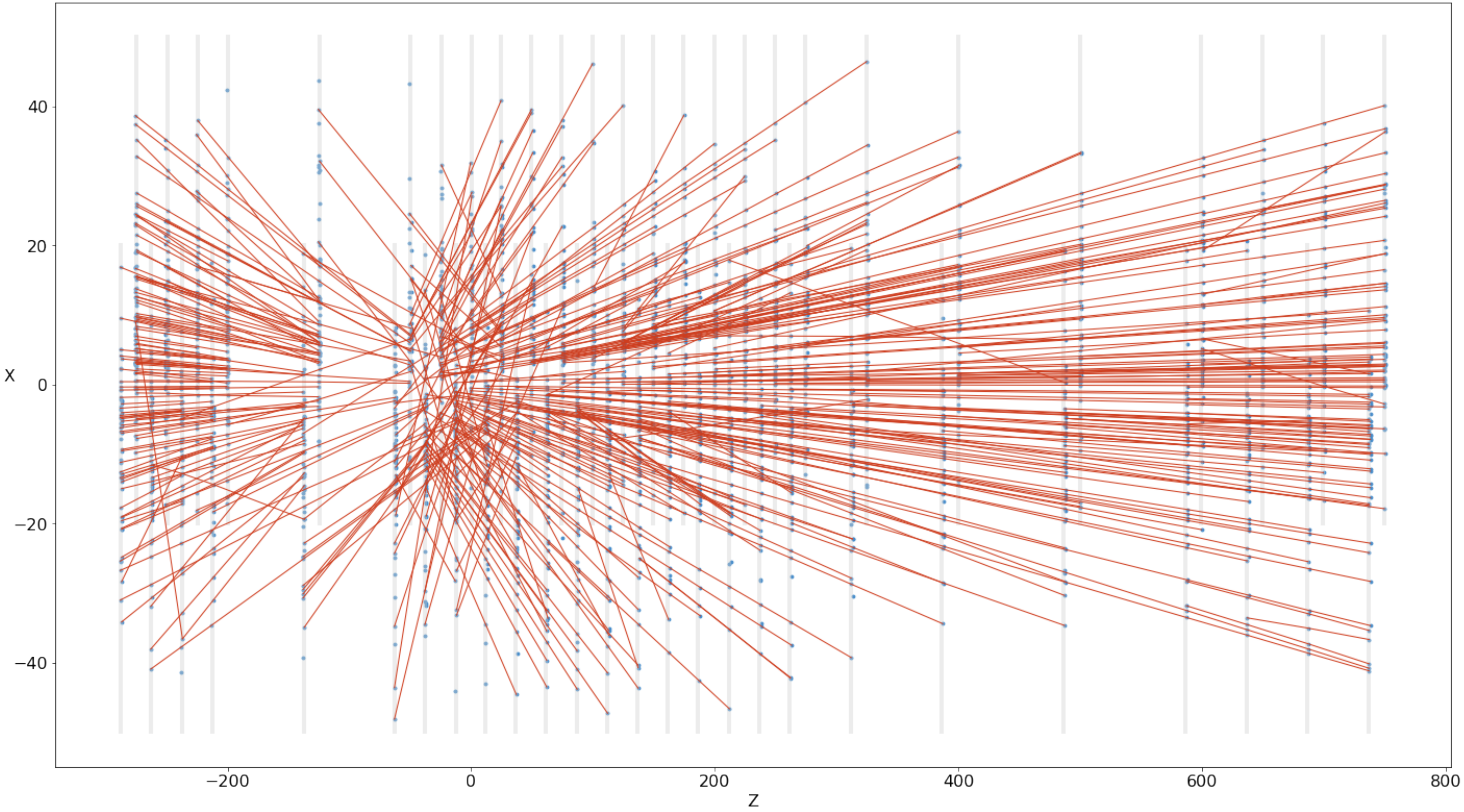
- The tracking system is composed of detectors: VELO, UT, SciFi (T stations).
- A magnetic field bends charged particles, we find out their **momentum**, **charge,** and the **collision / decay vertices** of the event

Going from this



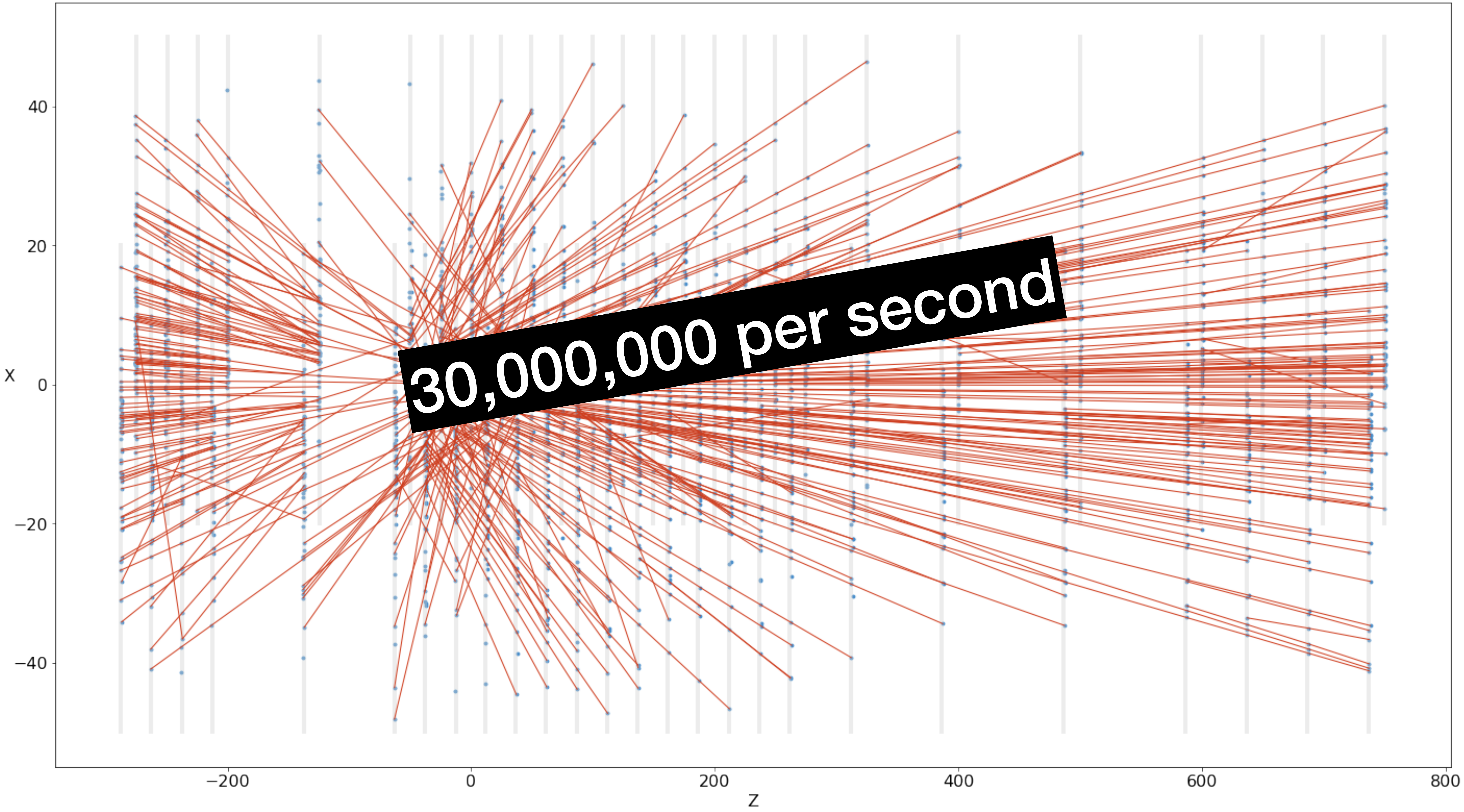
Daniel Cámpora

into this



Daniel Cámpora

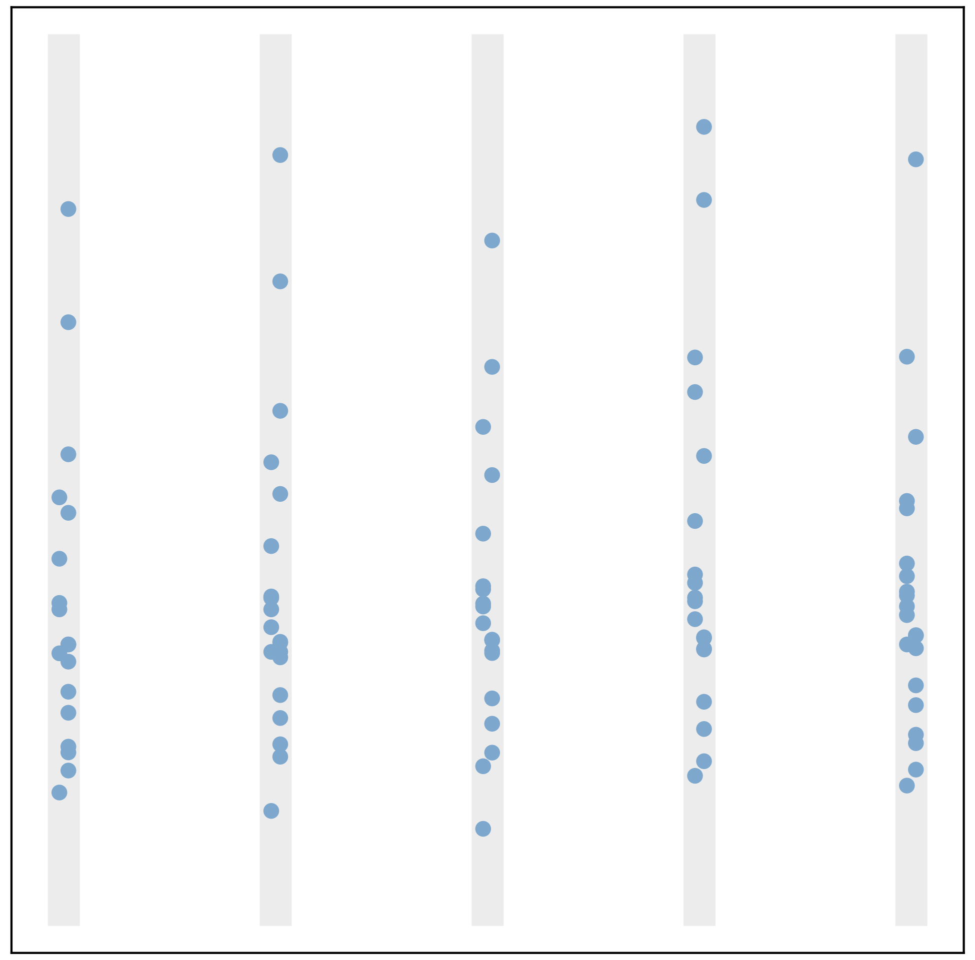
(at a high rate)

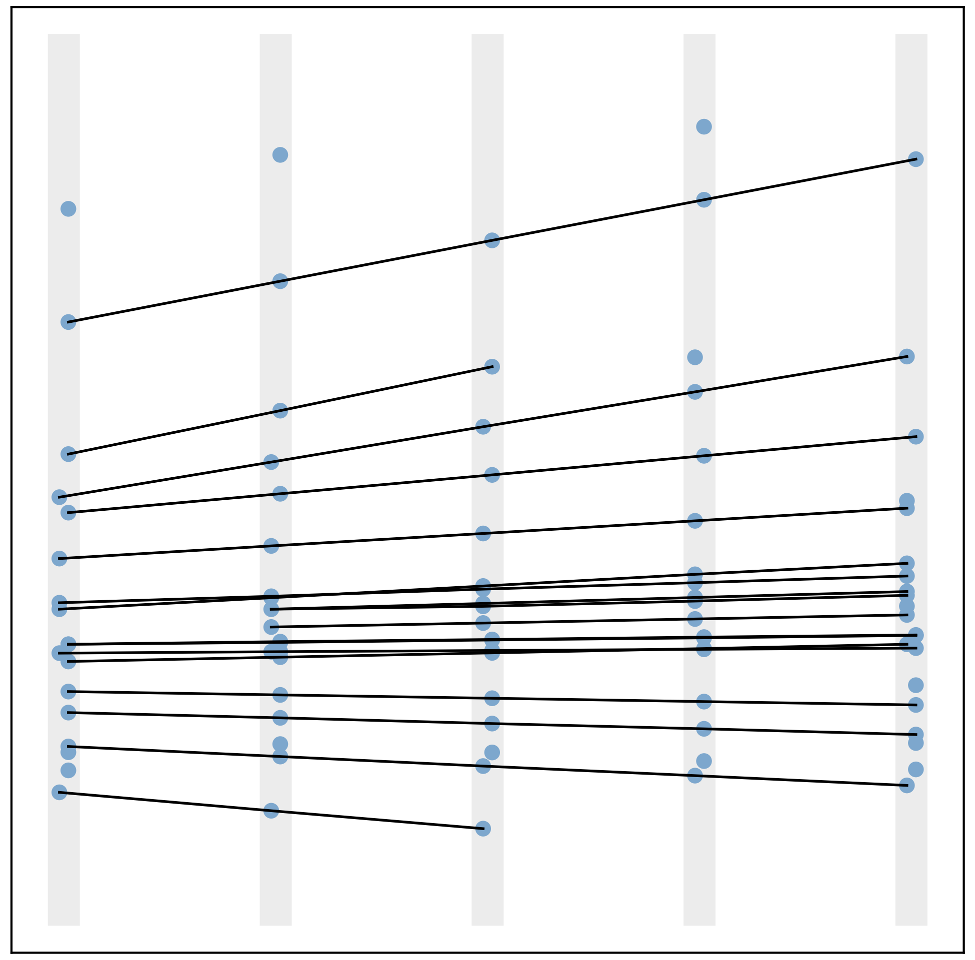


Daniel Cámpora

Taking VELO tracking as an example

- Rich literature of tracking methods
- Each event is physically independent
- Each track is independent
- Tracks come from a collision or secondary vertex
- VELO tracks are straight lines
- There is a geometrical distribution of hits across modules

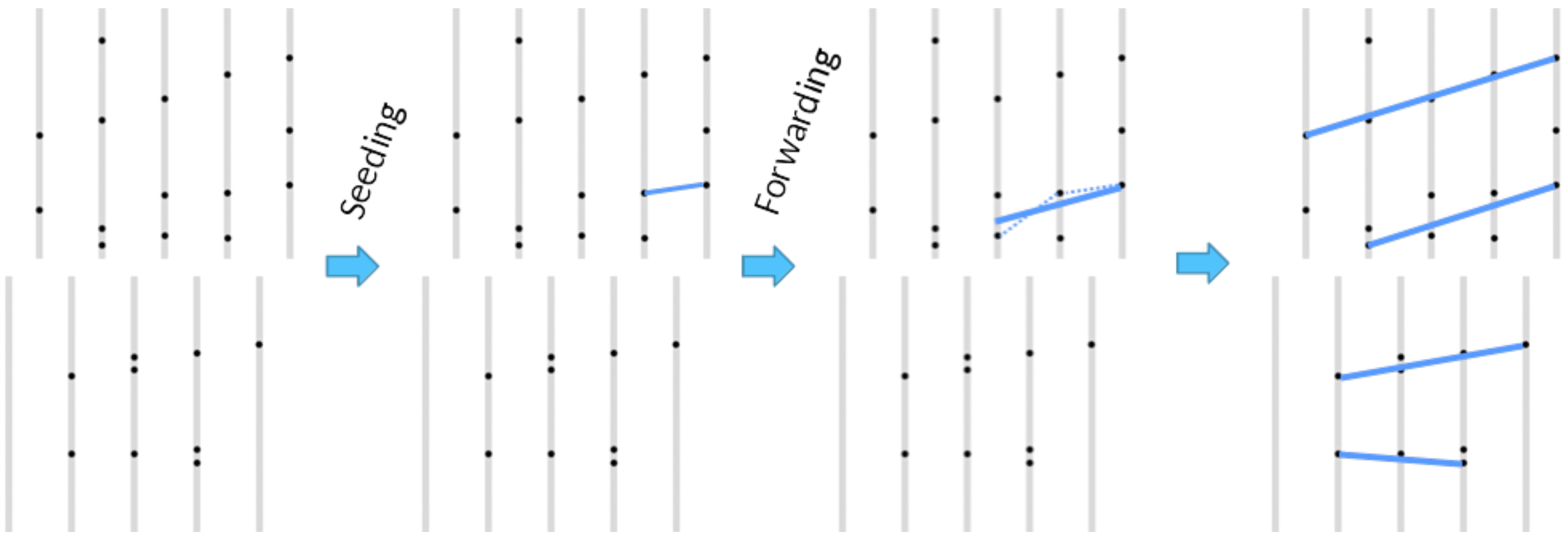




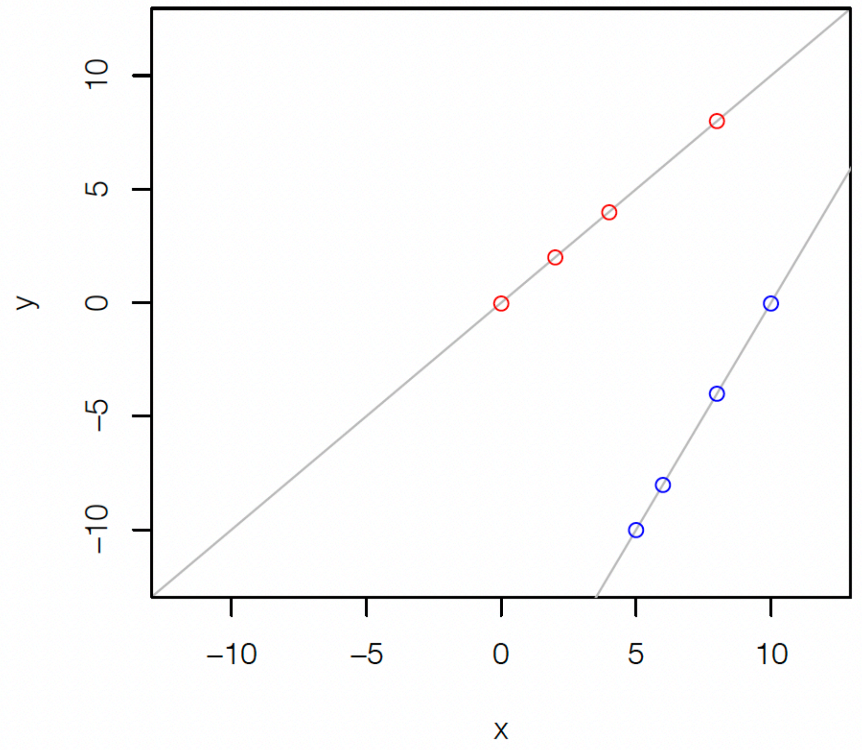
Daniel Cámpora

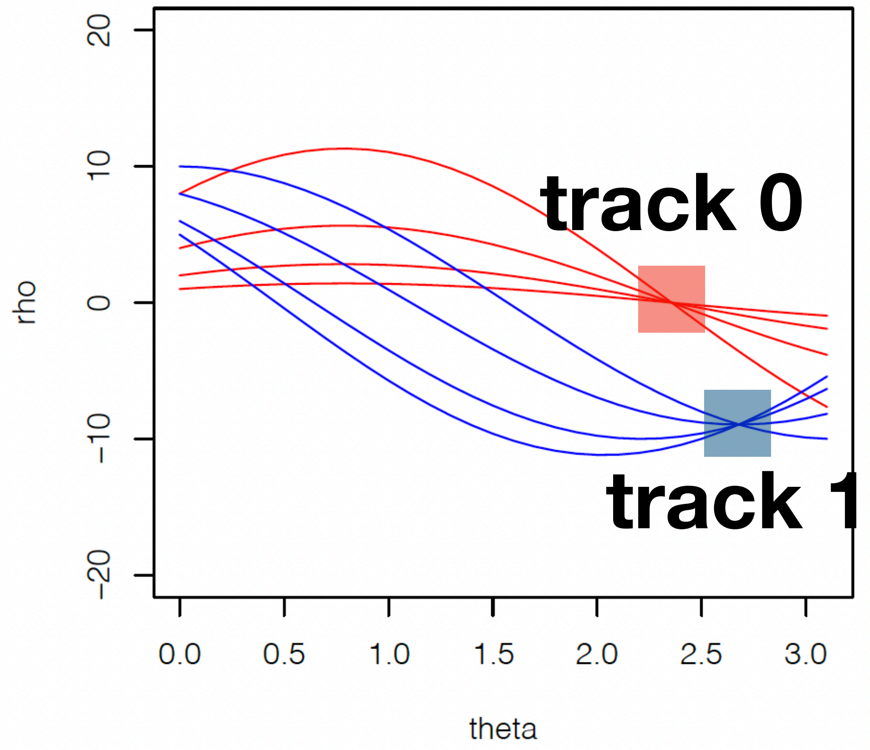
Rich literature of tracking methods

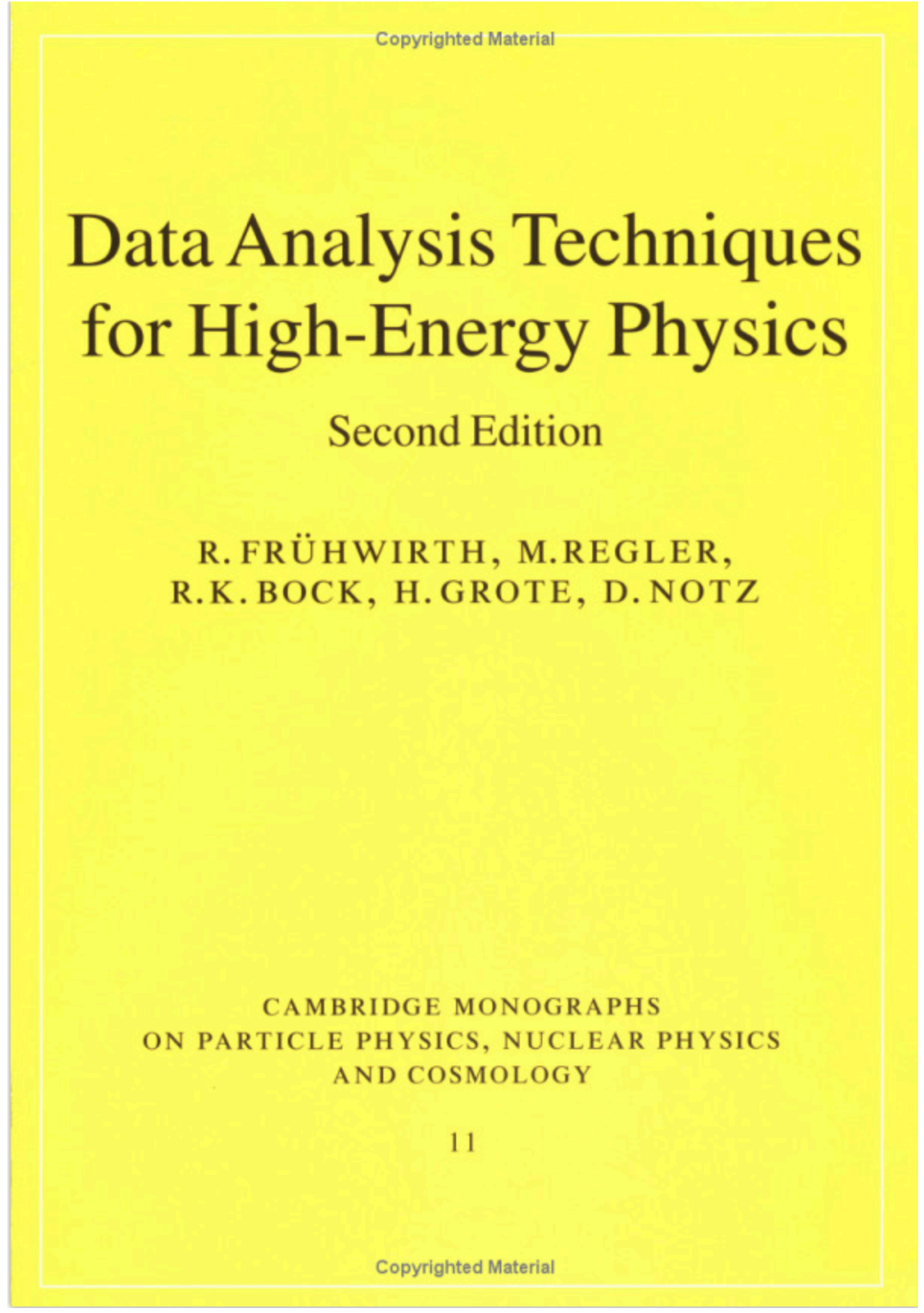
Local methods iteratively build a track and *follow* it:



Global methods map the problem to other equivalent formulations:

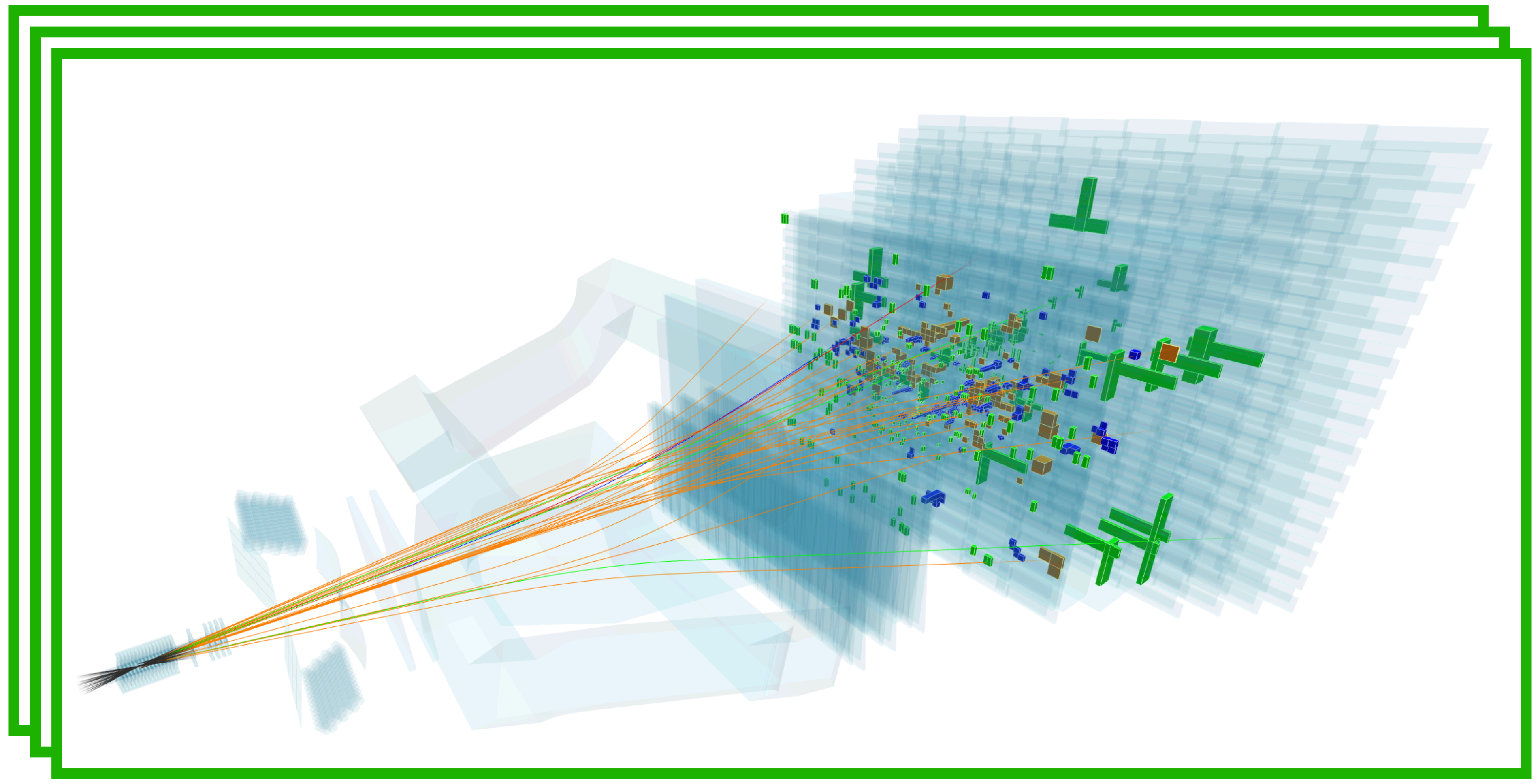






Each event is physically independent

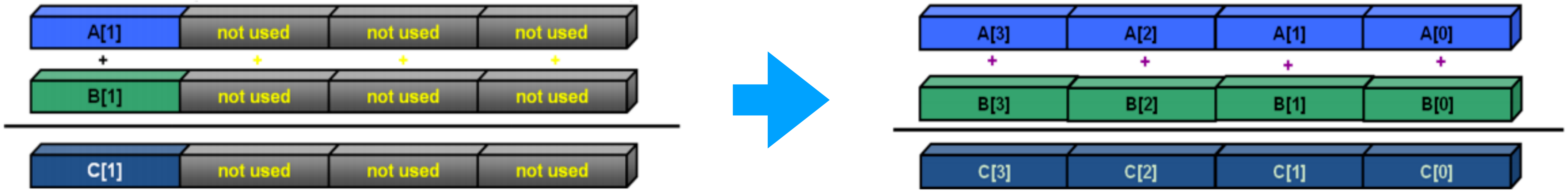
- We can therefore process them in separate CPU threads (as Gaudi does).
- On GPU, each event would be processed by a different *block.*
    - *Each block instantiates a different program, which is executed and managed by its own scheduler.*



Daniel Cámpora

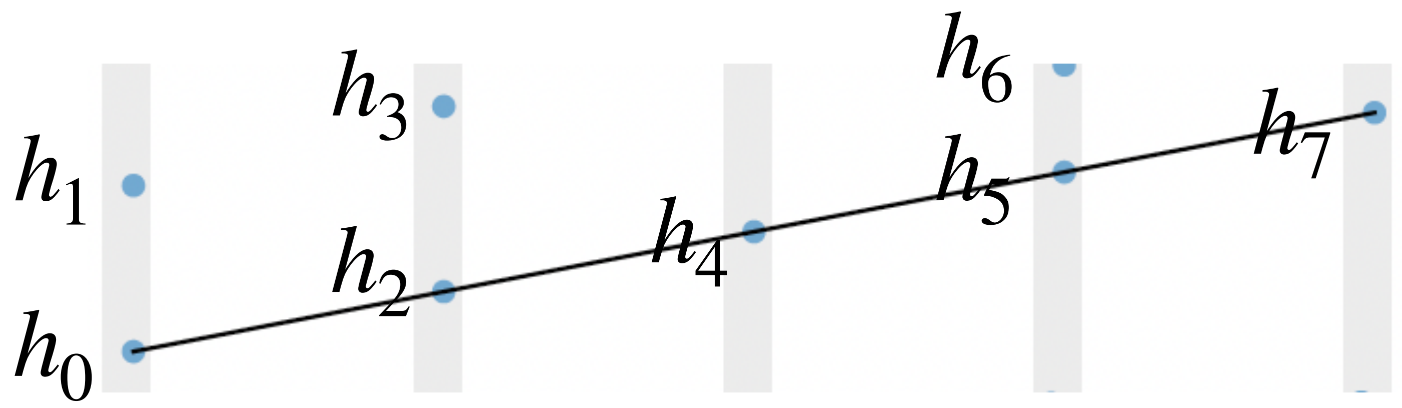
Each track is independent

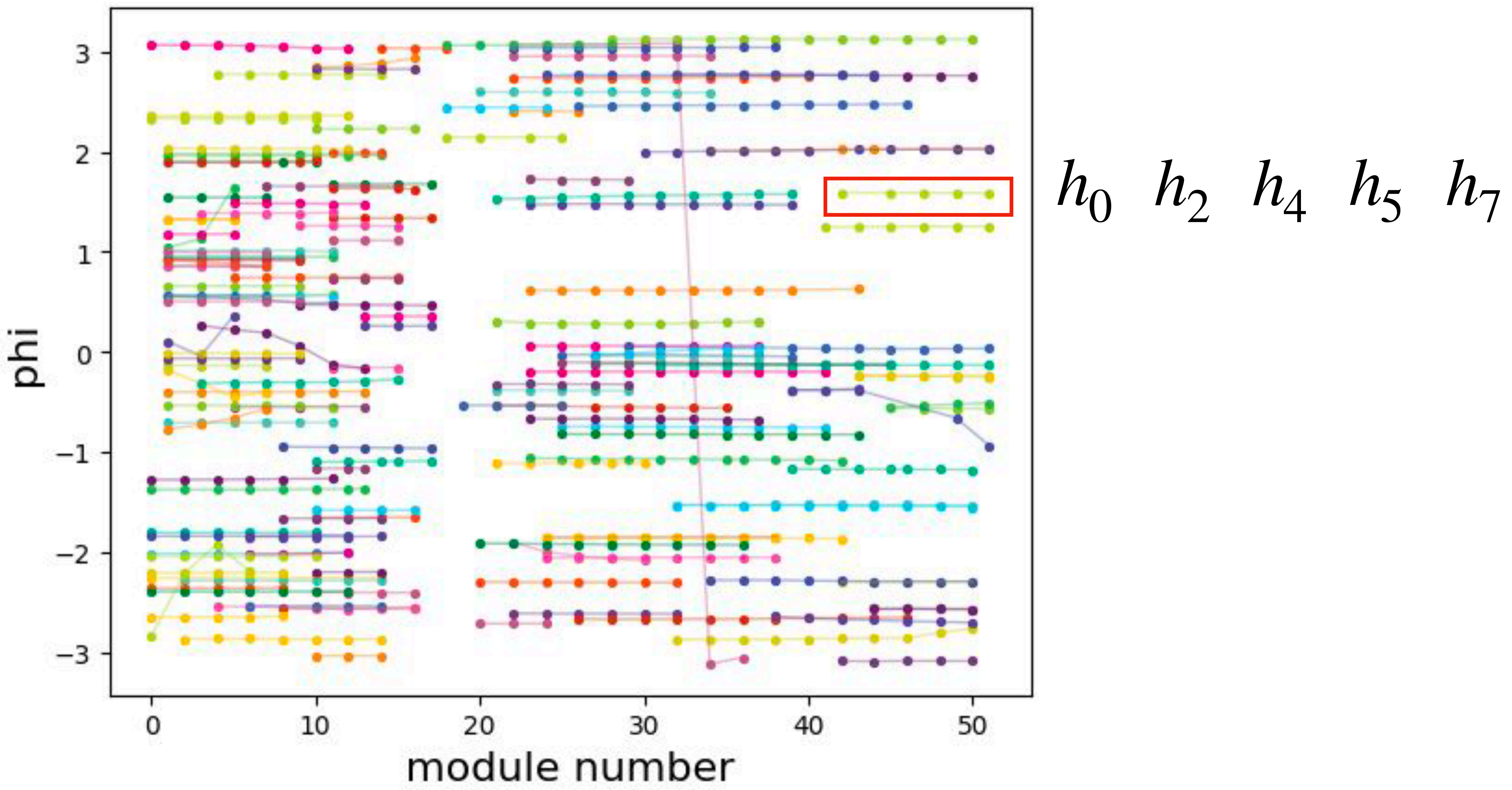
- Each thread can perform *Single-Instruction Multiple-Data* (SIMD) operations, processing several tracks in one go.
  - Modern CPUs have *SSE* and *AVX* extensions, leading to 4 or 8 FP32 simultaneous operations
  - GPUs use *warps* to control 32 threads at the same time
  - Operations must be homogeneous to take advantage



Tracks come from vertices

- We can exploit geometrical properties from the tracks to prepare an efficient data-structure.
  - In this case, sorting by phi is a good idea
  - Multi-dimensional structures (think 4D tracking) also exist

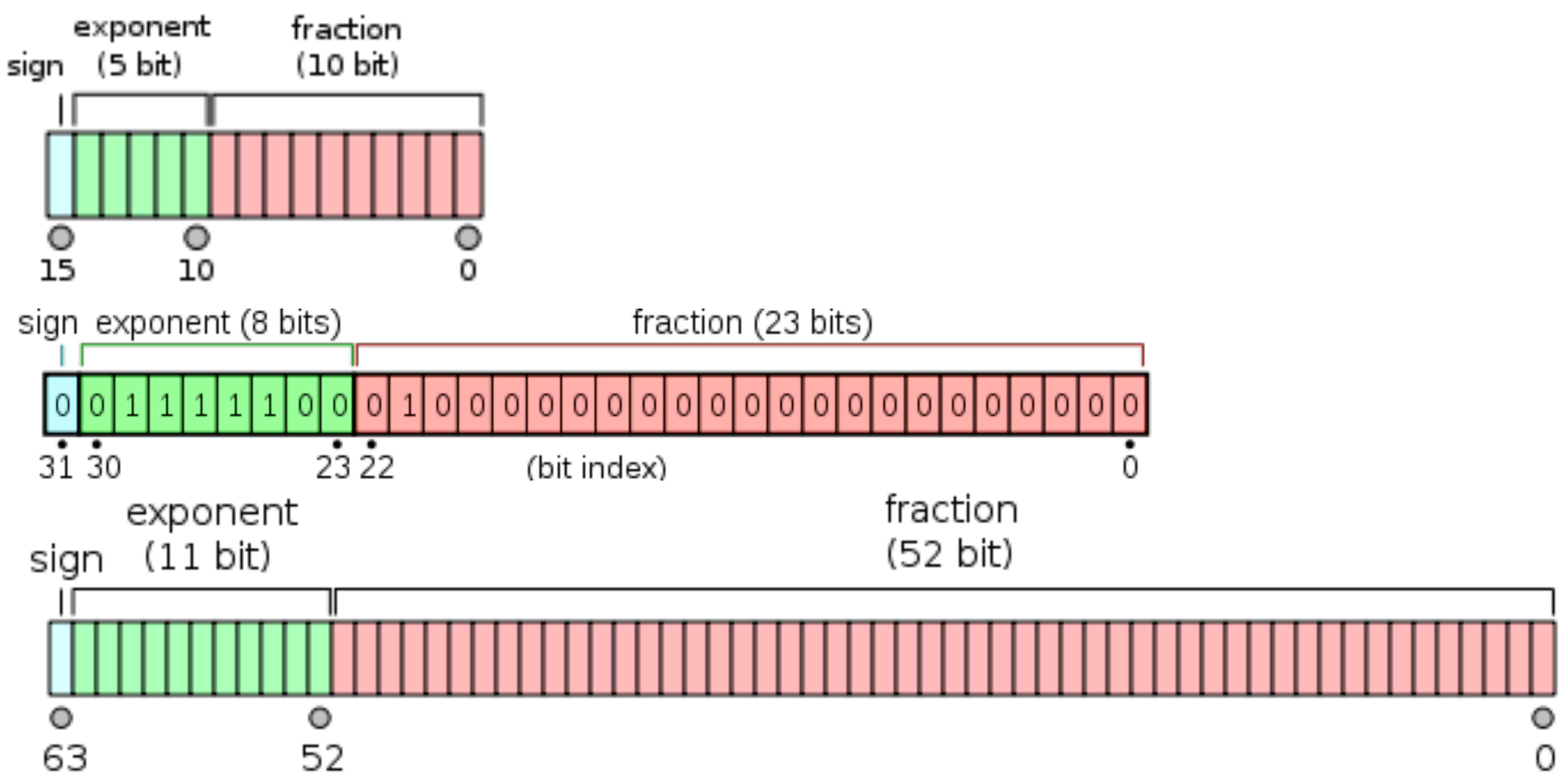




Daniel Cámpora

VELO tracks are straight lines

- The model for these tracks is not complex
- Use the necessary precision for your problem. The lower the better.
  - Distinguish between **arithmetic** and **storage**



Precision ↑ / Speed ↓

Arithmetic	Storage
Double precision	Double precision
Double precision	Single precision
Single precision	Single precision
Single precision	Half precision
Half precision	Half precision

Daniel Cámpora

Little exercise:
What's wrong with this piece of code?

```
__global__ void shared_memory_example(float* dev_array) {
  for (int i = threadIdx.x; i < 256; i += blockDim.x) {
    dev_array[i] = 1 / std::sqrt(2. + dev_array[i]);
  }
}
```

Little exercise (2)

```
__global__ void shared_memory_example(float* dev_array) {
  for (int i = threadIdx.x; i < 256; i += blockDim.x) {
    dev_array[i] = 1 / std::sqrt(2. + dev_array[i]);
  }
}
```

Use compiler flag -Wdouble-promotion to avoid surprises!

Little exercise (3)

```
__global__ void shared_memory_example(float* dev_array) {
  for (int i = threadIdx.x; i < 256; i += blockDim.x) {
    dev_array[i] = 1 / std::sqrt(2. + dev_array[i]);
  }
}
```

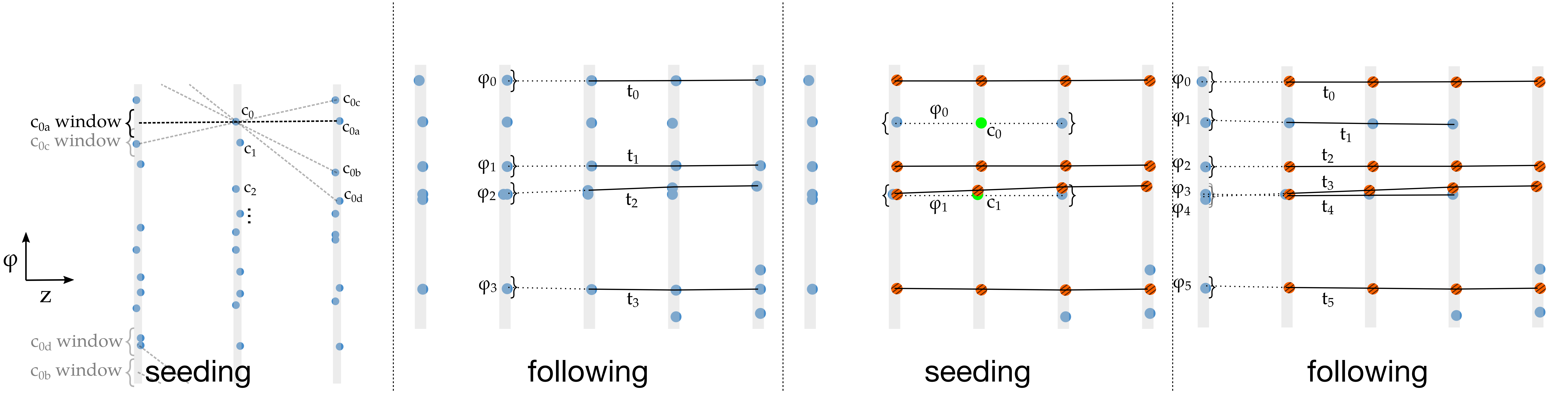
Use compiler flag -Wdouble-promotion to avoid surprises!

... and come to the thematic CERN School of Computing to learn more!

thematic CSC CERN School of Computing

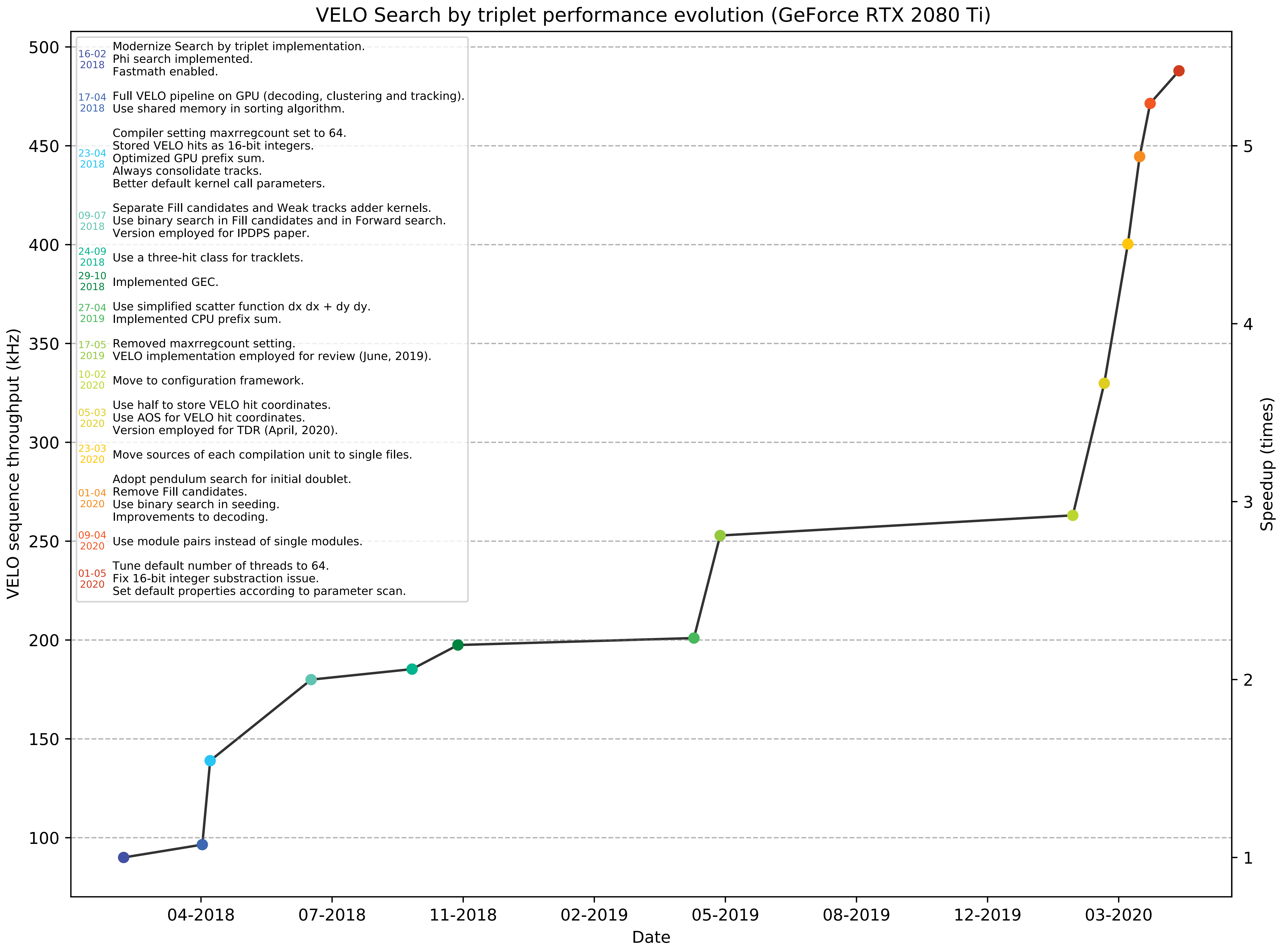
Geometrical properties

- The geometry of the detector should be used to find good access patterns
- Principle of locality:
  - **Spatial locality** - Prefer to access neighbouring data in memory
  - **Temporal locality** - Prefer reusing accessed data soon after first access



Daniel Cámpora

Applying all these principles

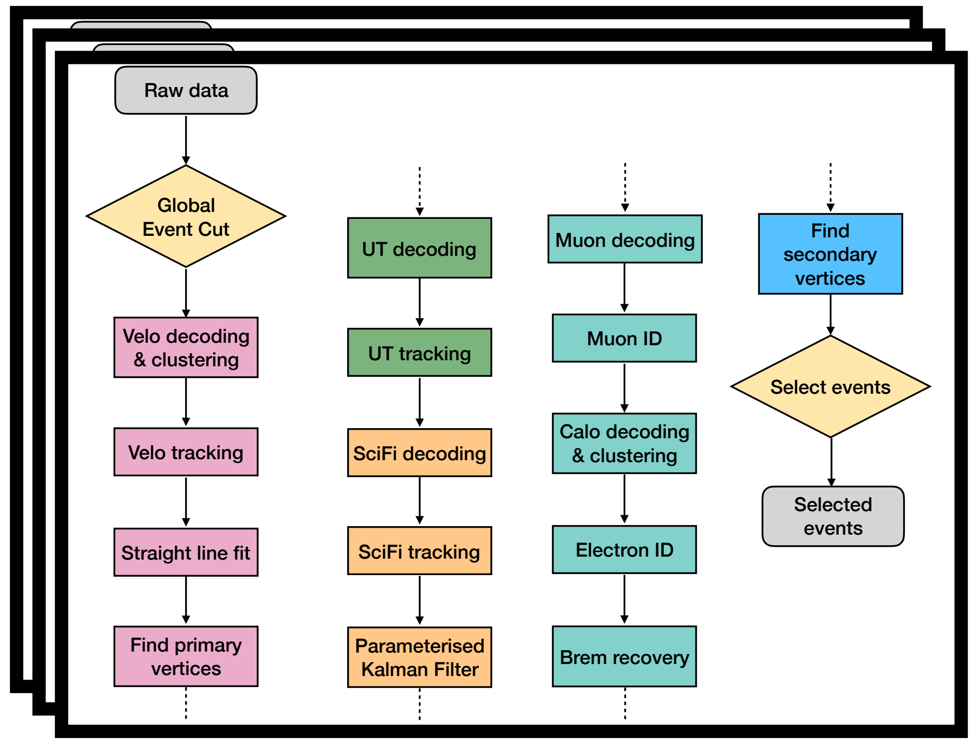


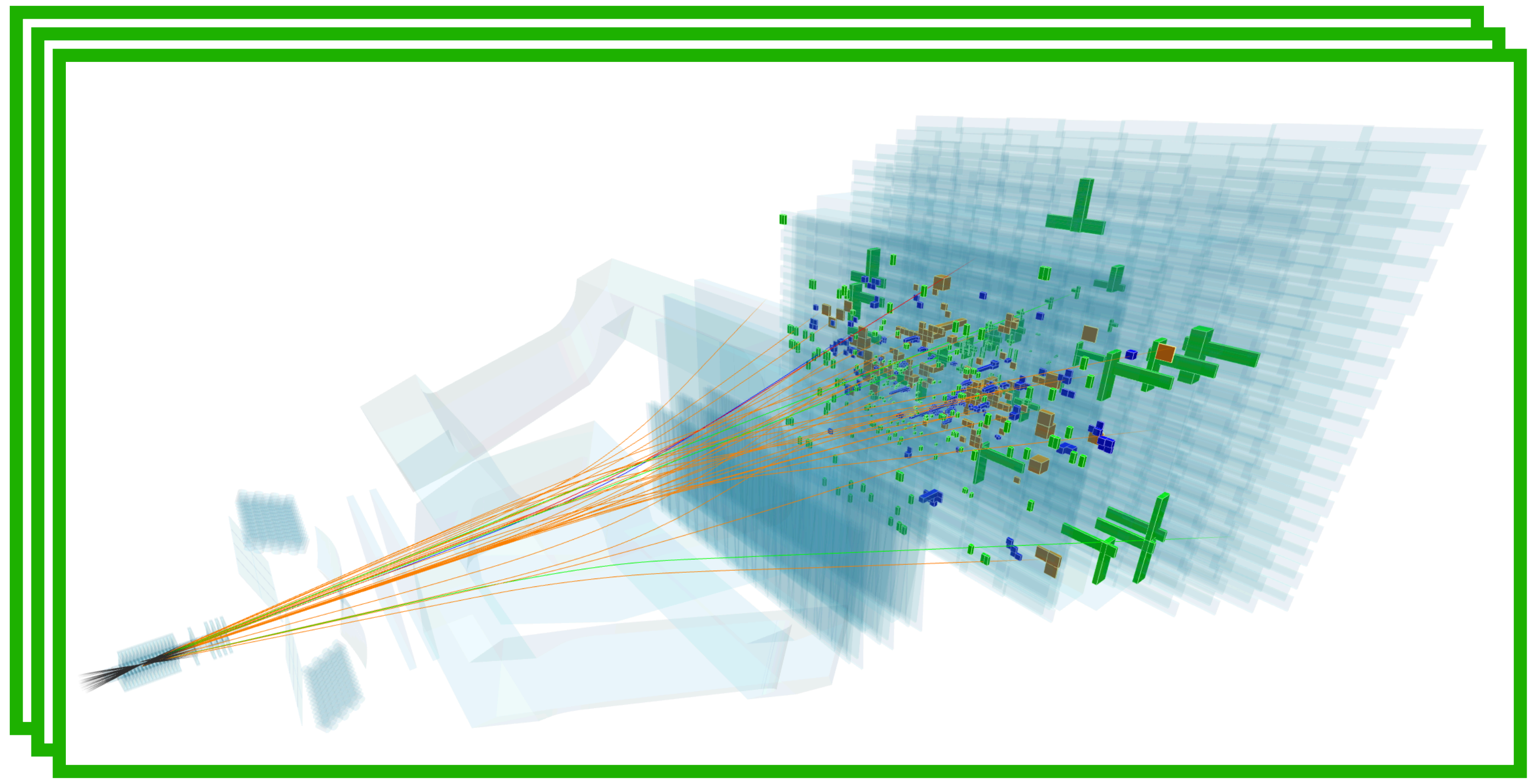
Daniel Cámpora

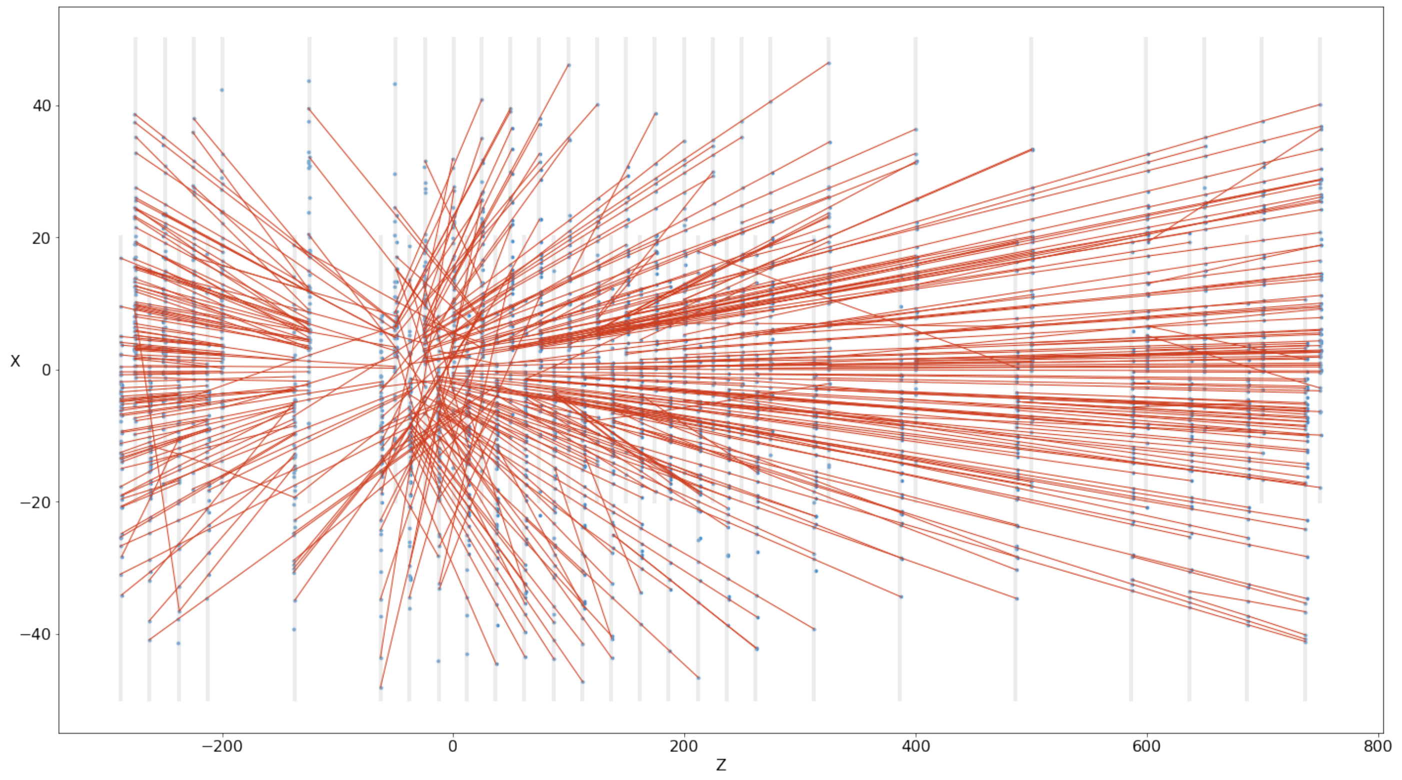
Parallelization

We parallelize at three levels:

- Sequences
- Events
- Intra-Algorithms

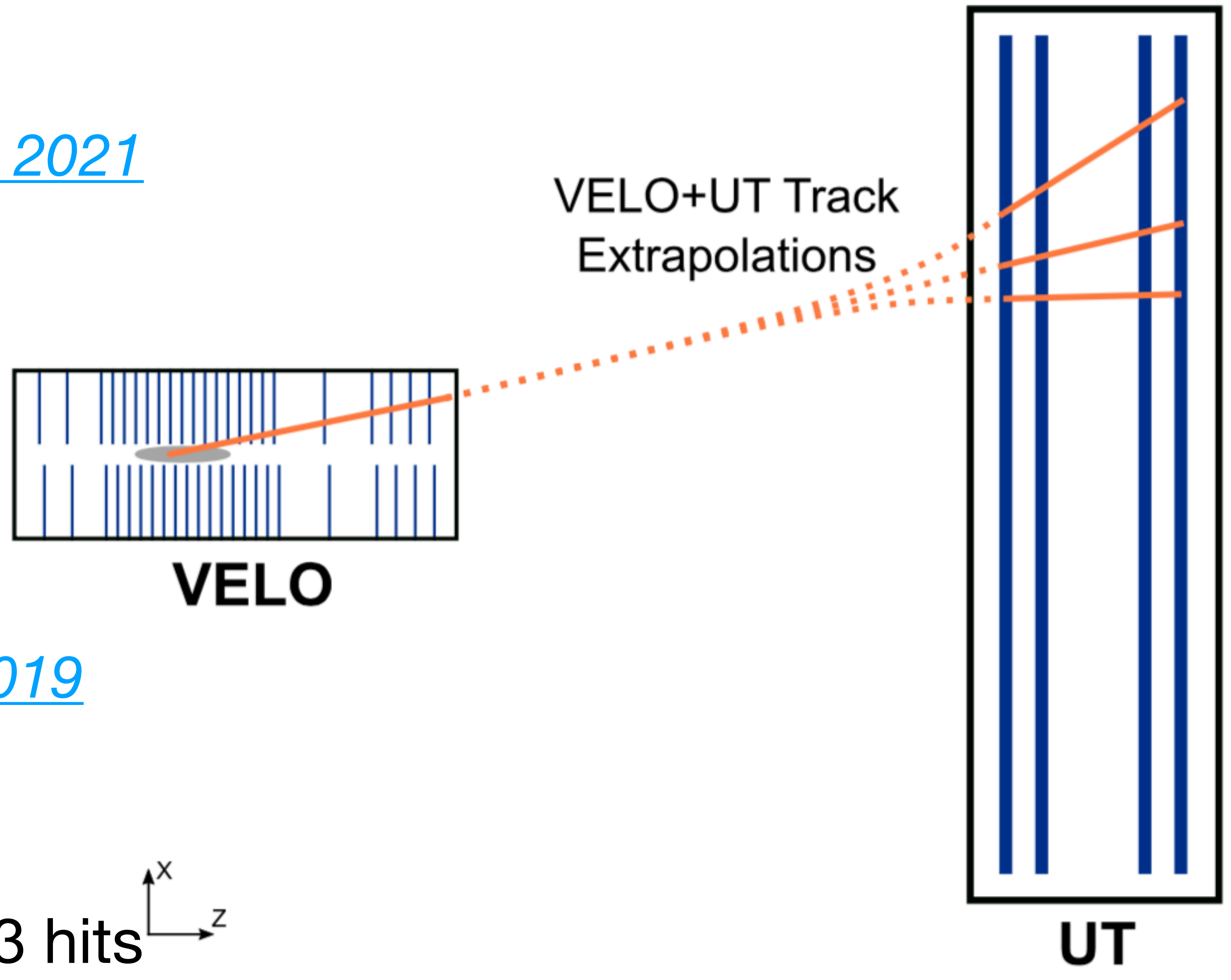






	Sequences	Events	Intra-Algorithms
CPU		Threads	Vectorisation
GPU	Streams	Blocks	Threads

Track reconstruction at HLT1

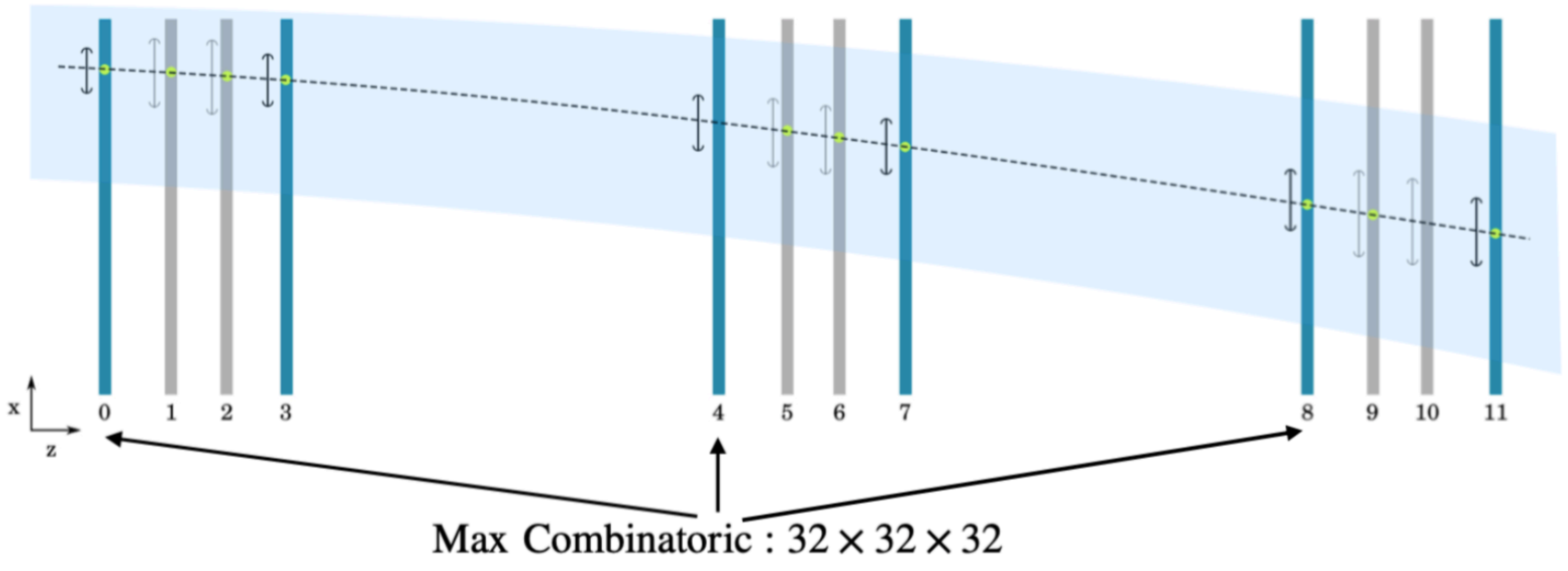


Velo tracking: [*Journal of Computational Science, vol. 54, 2021*]

- 52 silicon pixel modules with $\sigma_{x,y} \sim 5\ \mu m$
- Parallel local tracking algorithm: *Search by Triplet*
- Tracks fitted with simple Kalman filter assuming straight line model

Velo-UT tracking: [*IEEE Access, vol. 7, pp. 91612-91626, 2019*]

- 4 layers of silicon strips
- Velo tracks extrapolated to UT taking into account fringe B-field
- Parallelized tracklet finding inside search windows requiring at least 3 hits



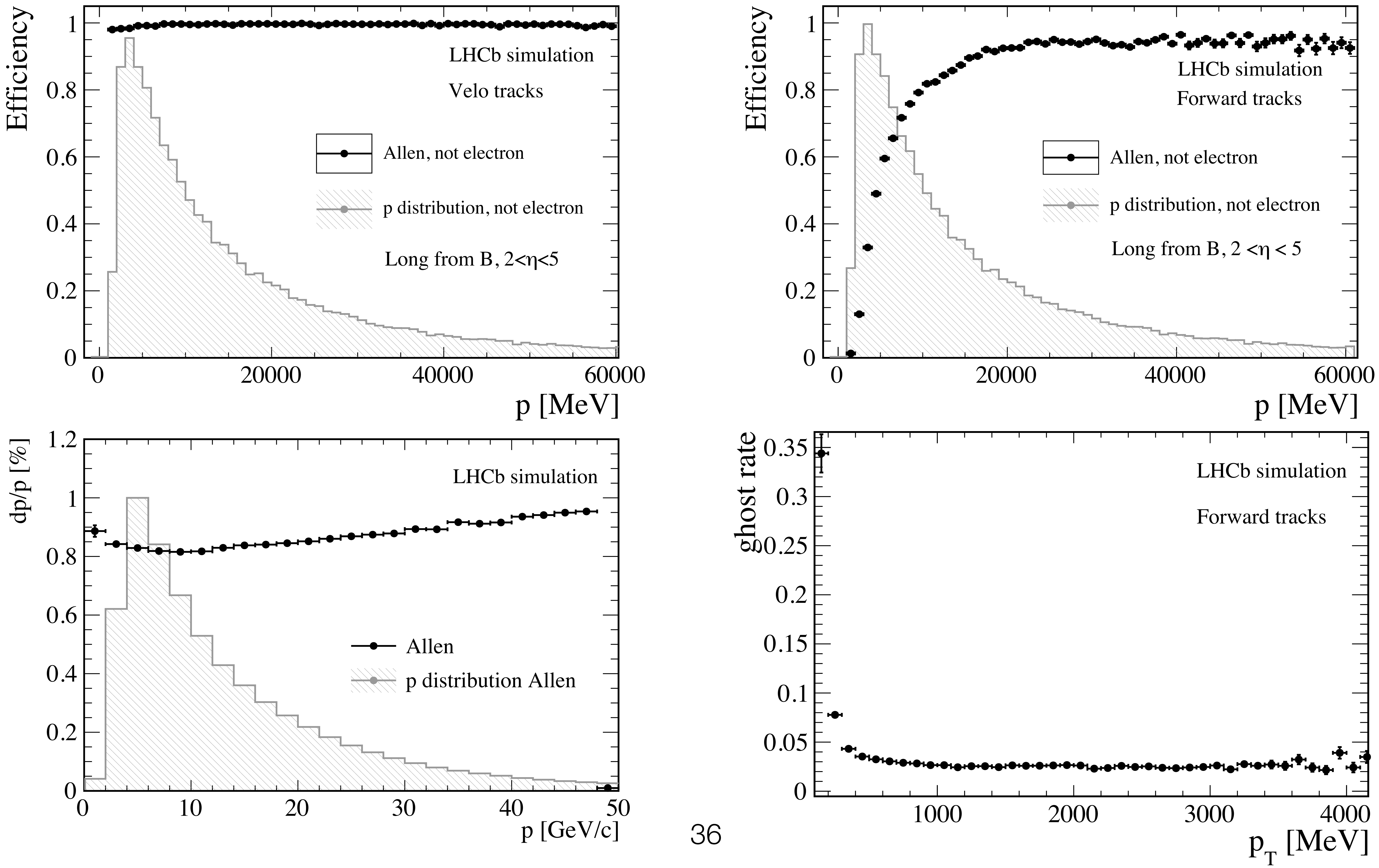
Forward tracking: [*Comput Softw Big Sci 4, 7 (2020)*]

- 3 stations with 4 layers of Scintillating Fibres
- Velo-UT tracks extrapolated using parametrization
- Parallelized *Forward algorithm* to reconstruct **long tracks:**
  - Search windows from on Velo-UT momentum estimate
  - Form triplets and extend to remaining layers

Daniel Cámpora

HLT1 tracking efficiency

- **Run 2 efficiency maintained at x5 instantaneous luminosity**
- Excellent track reconstruction efficiency (> 99% for VELO, 95% for high-p forward tracks)
- Good momentum resolution and fake rejection

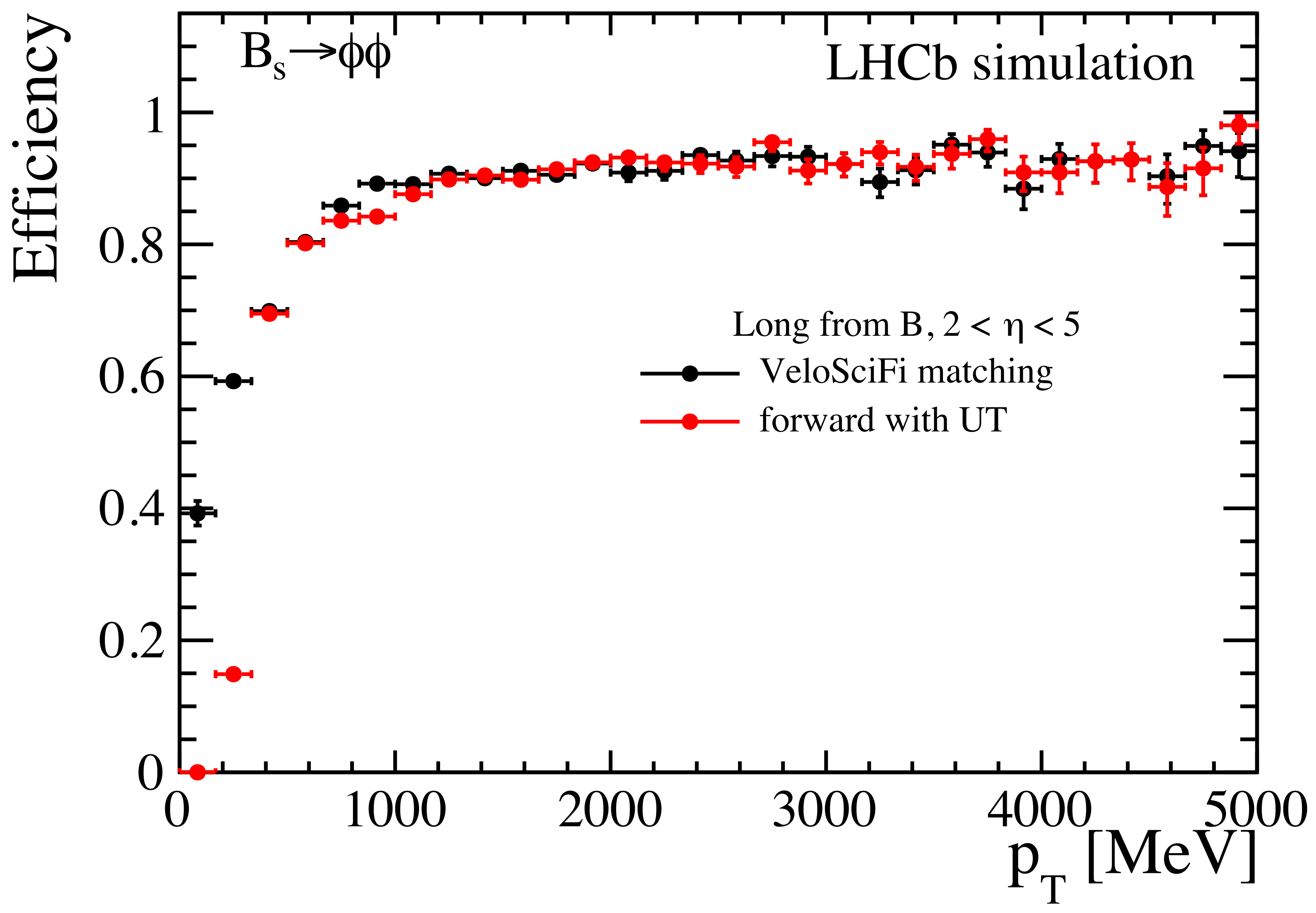


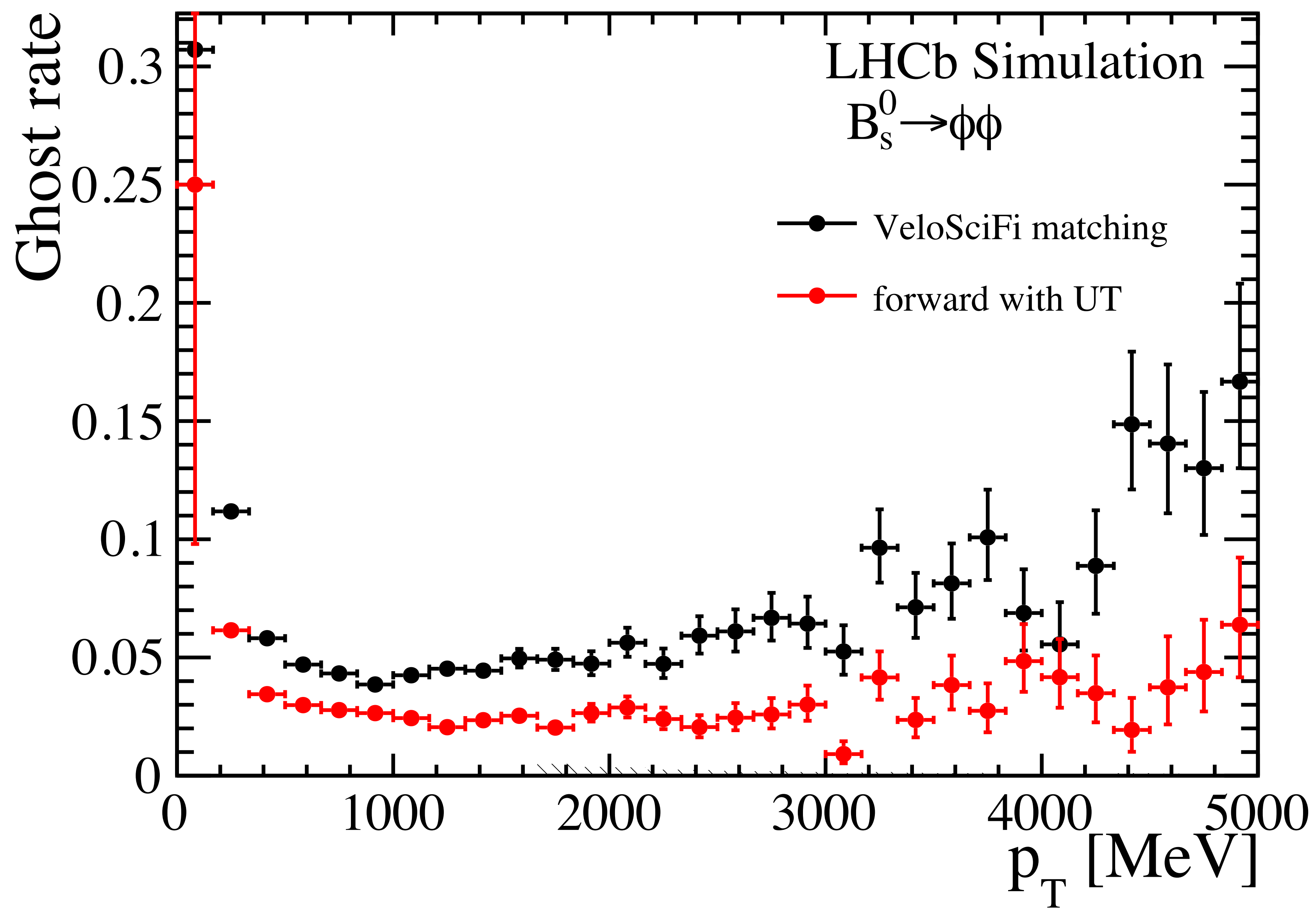
LHCb-FIGURE-2020-014

Daniel Cámpora

Tracking without the UT

- In 2022, the UT detector will unfortunately not be available for data-taking
- Tracking performance and throughput maintained, at the cost of larger fake rate
- Opportunity to commission 2 options, which **both maintain the current throughput:**
  - **Forward without UT**
  - **Seeding+Matching:**
    - Standalone SciFi reconstruction & matching to VELO seeds
    - Highly efficient for low momenta
    - Opens the door to additional physics cases in HLT1 (downstream and SciFi tracks)



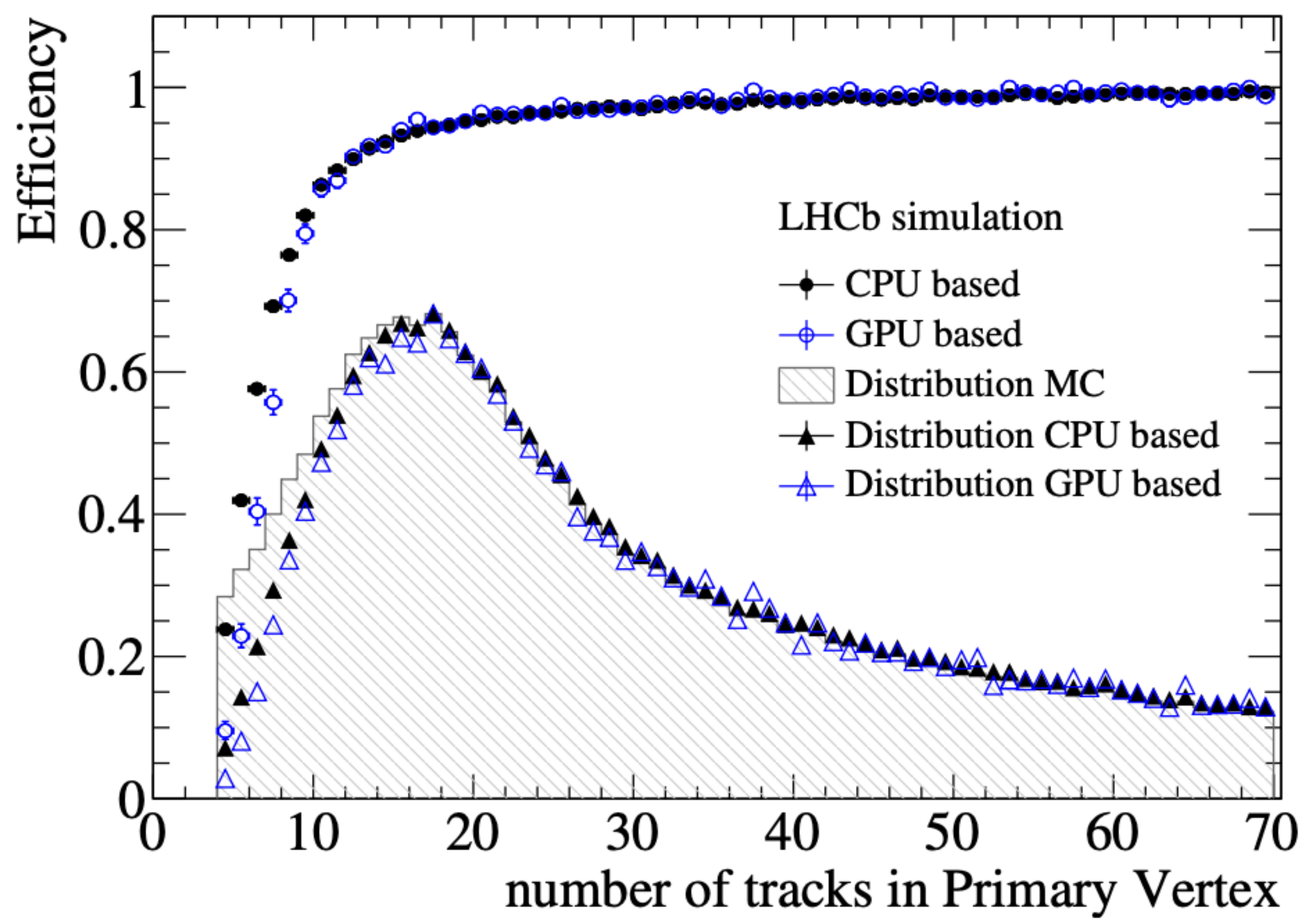


LHCB-FIGURE-2022-010

Daniel Cámpora

Vertex reconstruction

- Primary vertices found from **clusters** in the closest approach of tracks to the beamline
- 1-1 mapping between tracks and vertices requires **serialization**
  - Instead, every track assigned to every vertex based on **weight**
- **Efficiency > 90%** for vertices with N. tracks > 10



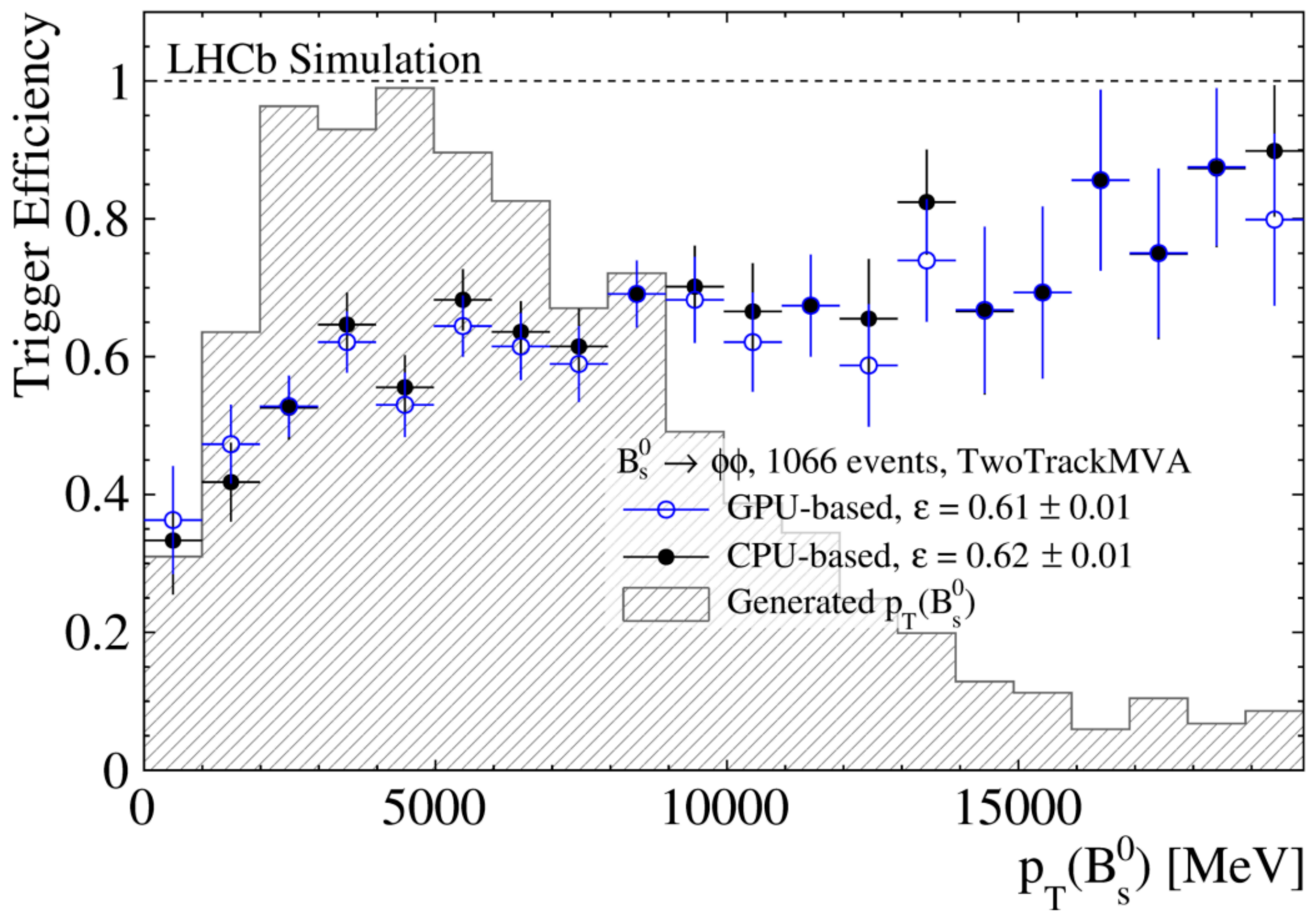
[Comput. Softw. Big Sci. 6 (2022) no.1, 1](#)

Daniel Cámpora

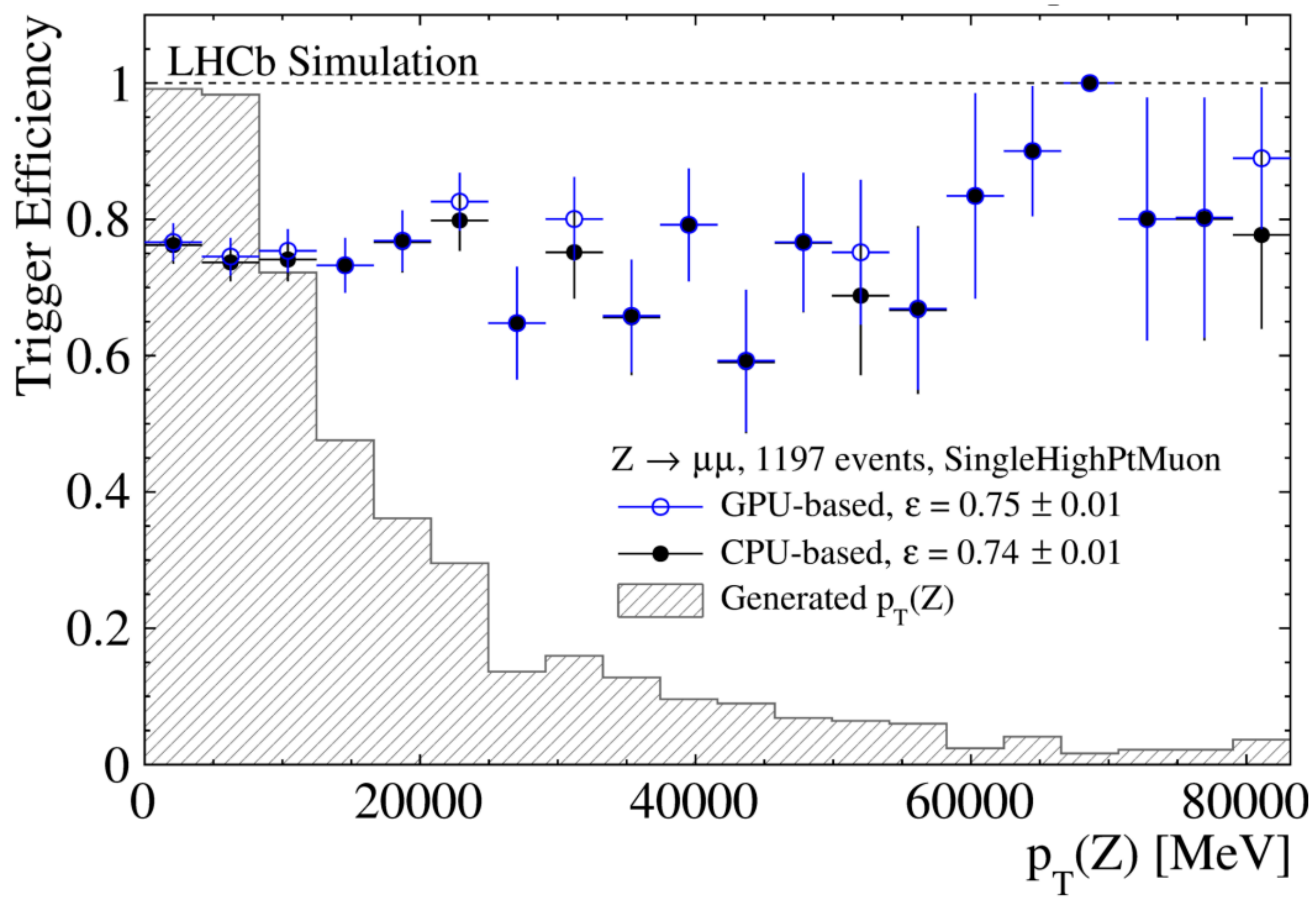
HLT1 selection performance

- Inclusive rate for the main HLT1 lines ~ 1 MHz
- O(30) lines implemented so far:
  - Cover majority of LHCb physics program (B, D decays, semileptonic, EW physics)
  - Special lines for monitoring, alignment and calibration
  - Additional trigger lines under development
  - Compatible performance between CPU and GPU

TwoTrackMVA line for $B_s^0 \to \phi\phi$



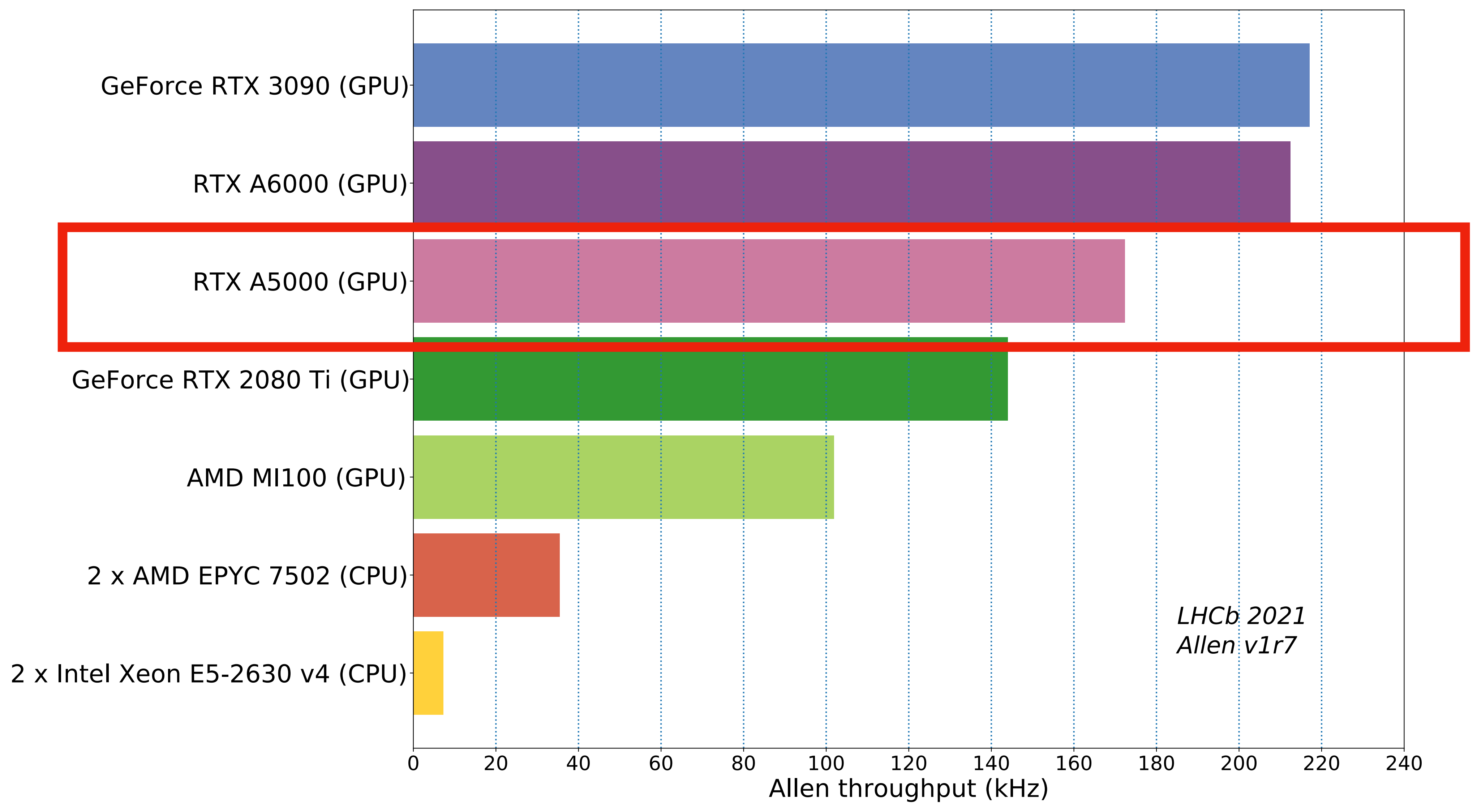
SingleHighPtMuon line for $Z \to \mu^+\mu^-$

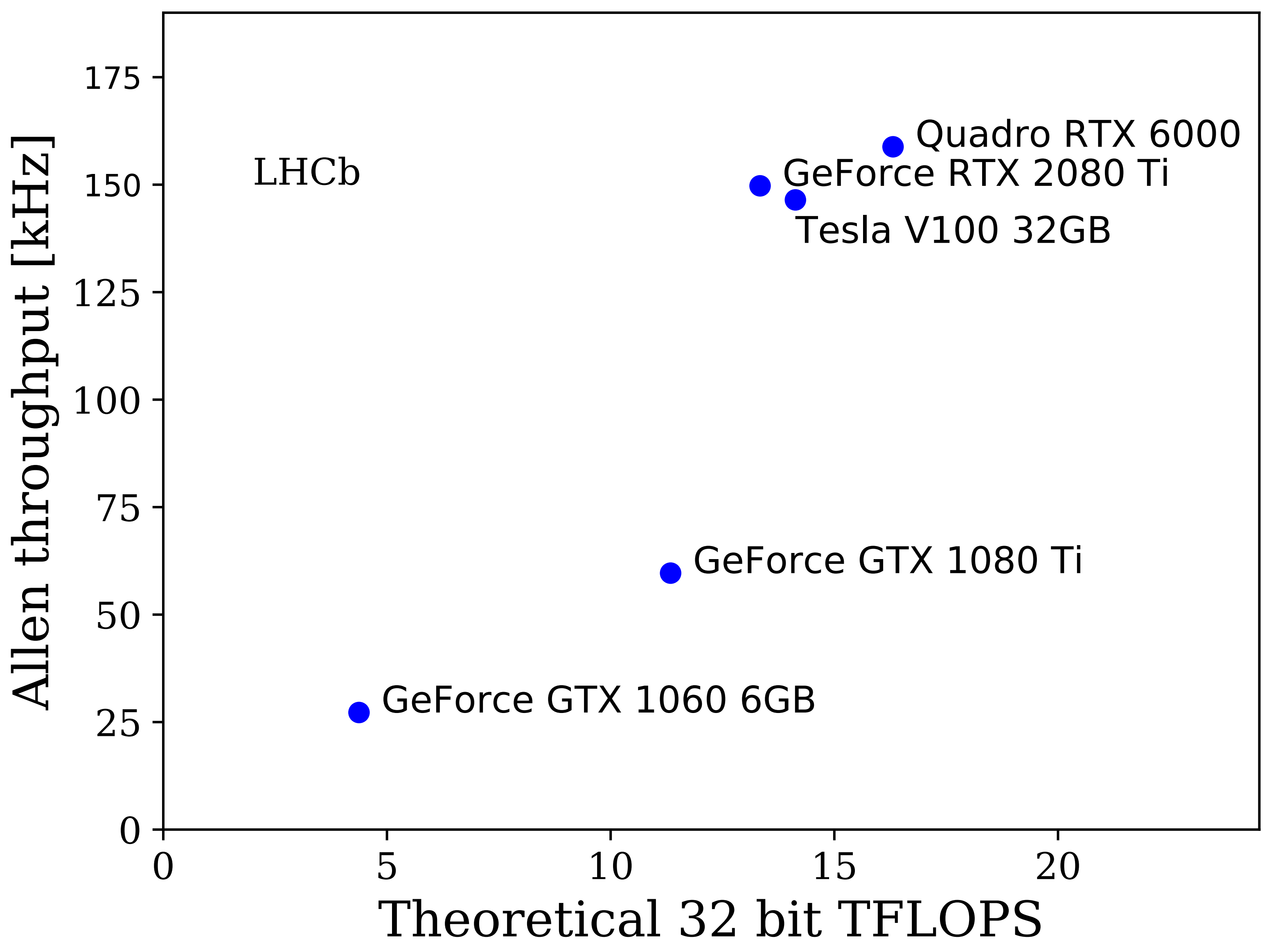


Comput. Softw. Big Sci. 6 (2022) no.1, 1

Throughput

- 30 MHz goal can be achieved with O(200) GPUs (maximum the Event Builder server can host is 500)
- Throughput scales well with theoretical TFLOPS of GPU card
- Additional functionalities are being explored





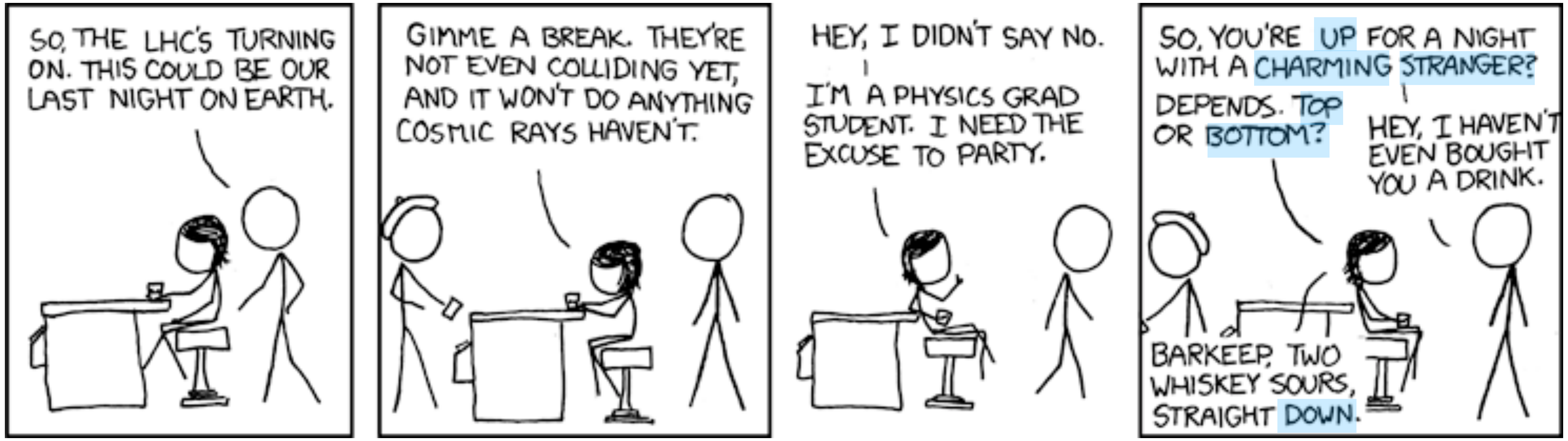
LHCb-FIGURE-2020-014

Daniel Cámpora

About the hardware

About parallel algorithms

About commissioning



Previously...

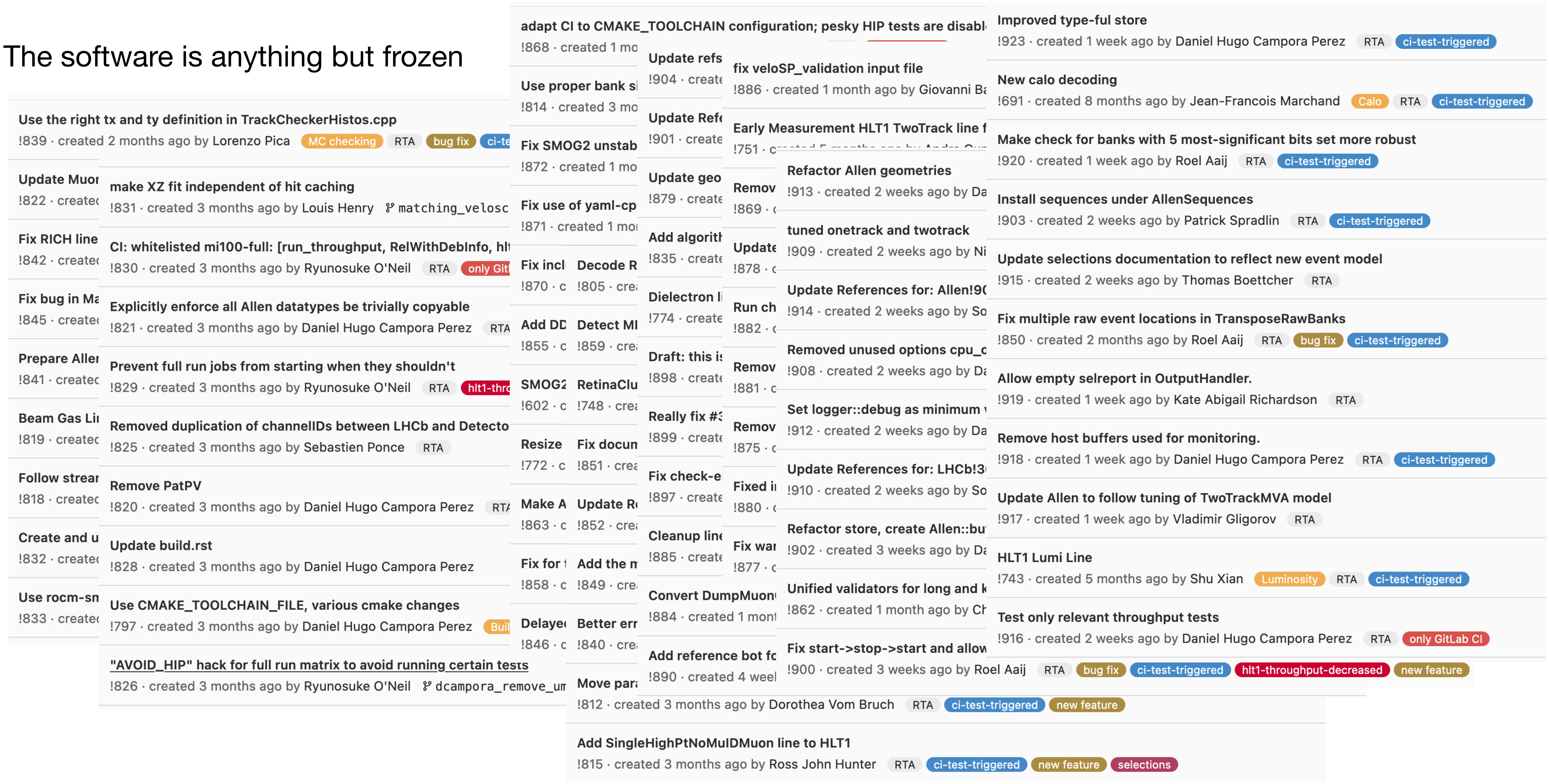
LHCb October beam-test was major milestone for the Allen commissioning

Steps for the future:

- As more sub-detectors get installed, commission more parts of the **decoding**, **reconstruction** and **selection** chain
- Commission the **full chain** (EB → HLT1 → HLT2 → storage & offline)
- **Monitoring**
- Continue the *installation of GPU cards* in the LHCb Data center
- **Throughput**, **memory**, **cooling** and **stability** tests with larger-scale system
- *Data taking* with stable beams expected to start in spring 2022!

Over the last months

The software is anything but frozen

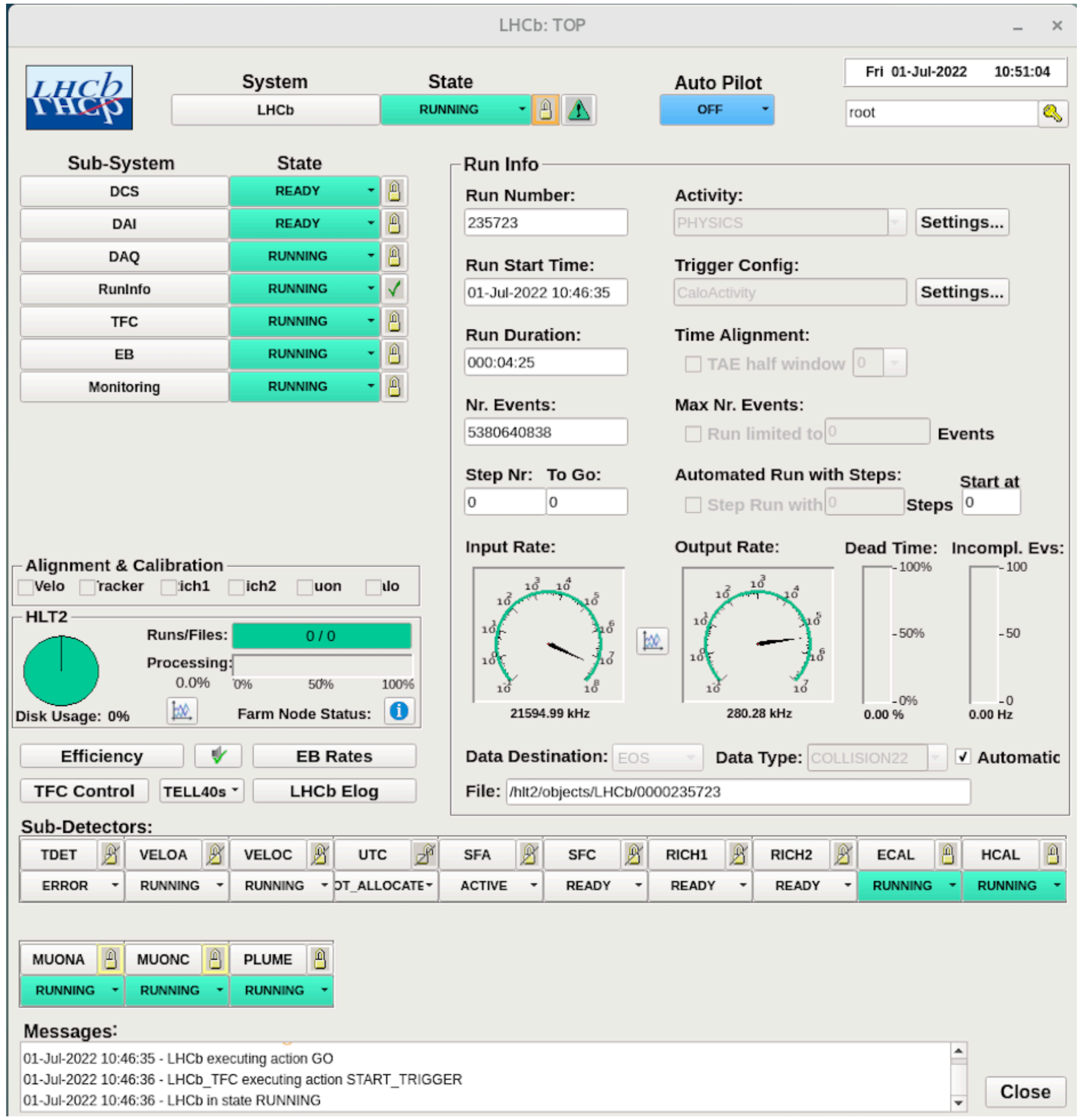


Daniel Cámpora

Commissioning is a changing objective

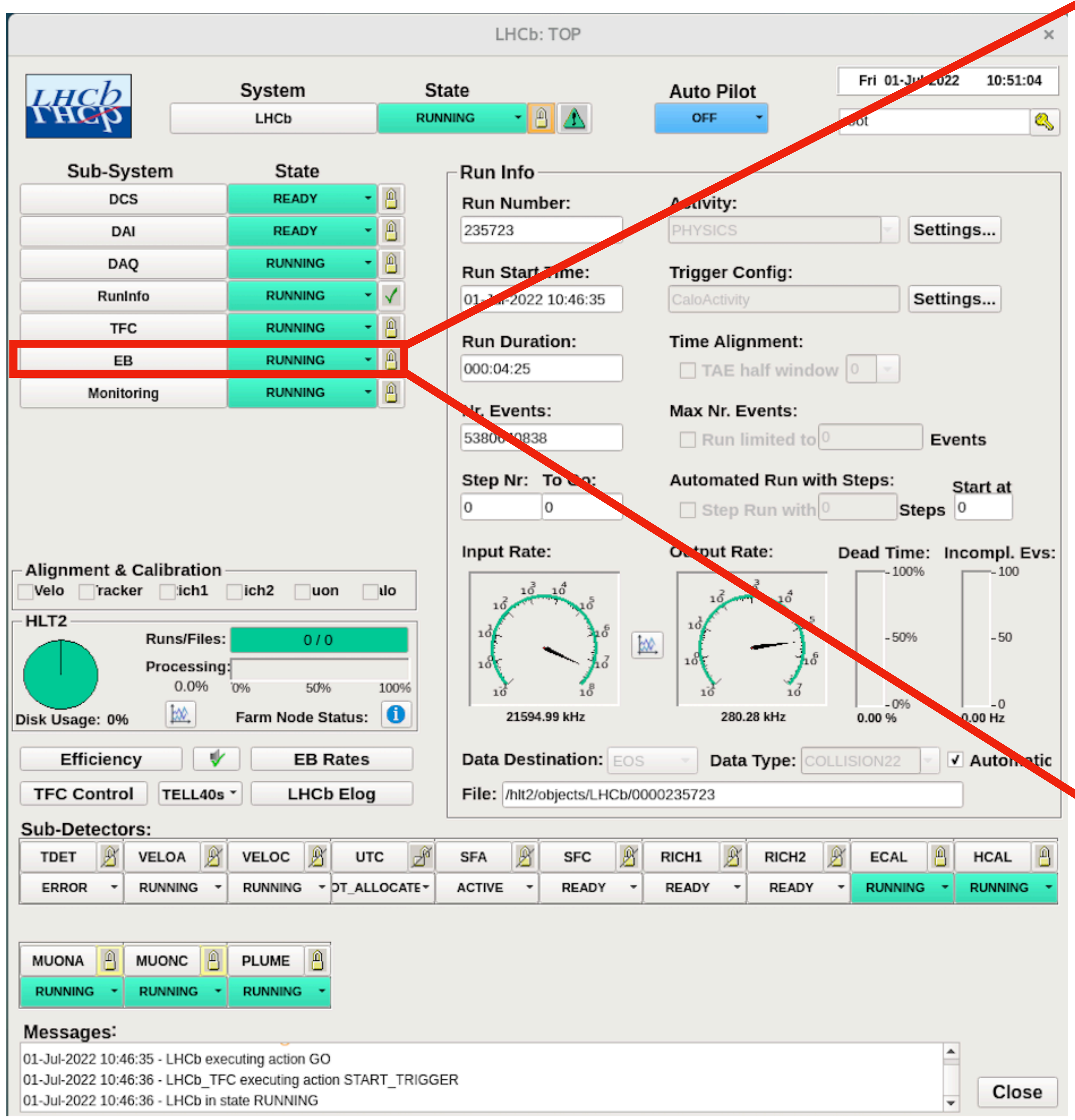
- Decoding status: ecal, muons, VELO, SciFi, plume (from now on dev for HLT1 and HLT2 simultaneous)
- Reconstruction status: Forward tracking, VELO-SciFi matching
- Selections: 30 lines, scalable (~10% throughput hit scaling to 100 lines)
- Monitoring: progressing well
- Throughput / stability tests: Up to 30 MHz without any dropped packets (*full chain*, calo clustering or passthrough 1/25). One can push up to 40 MHz, *Allen is not the bottleneck*
- All GPUs are installed, operational and tested (also all FPGAs, EB nodes, etc.)
- In general, more monitoring is needed to identify issues more quickly
- Advances every week: **excellent work atmosphere, good pace, responsive team**

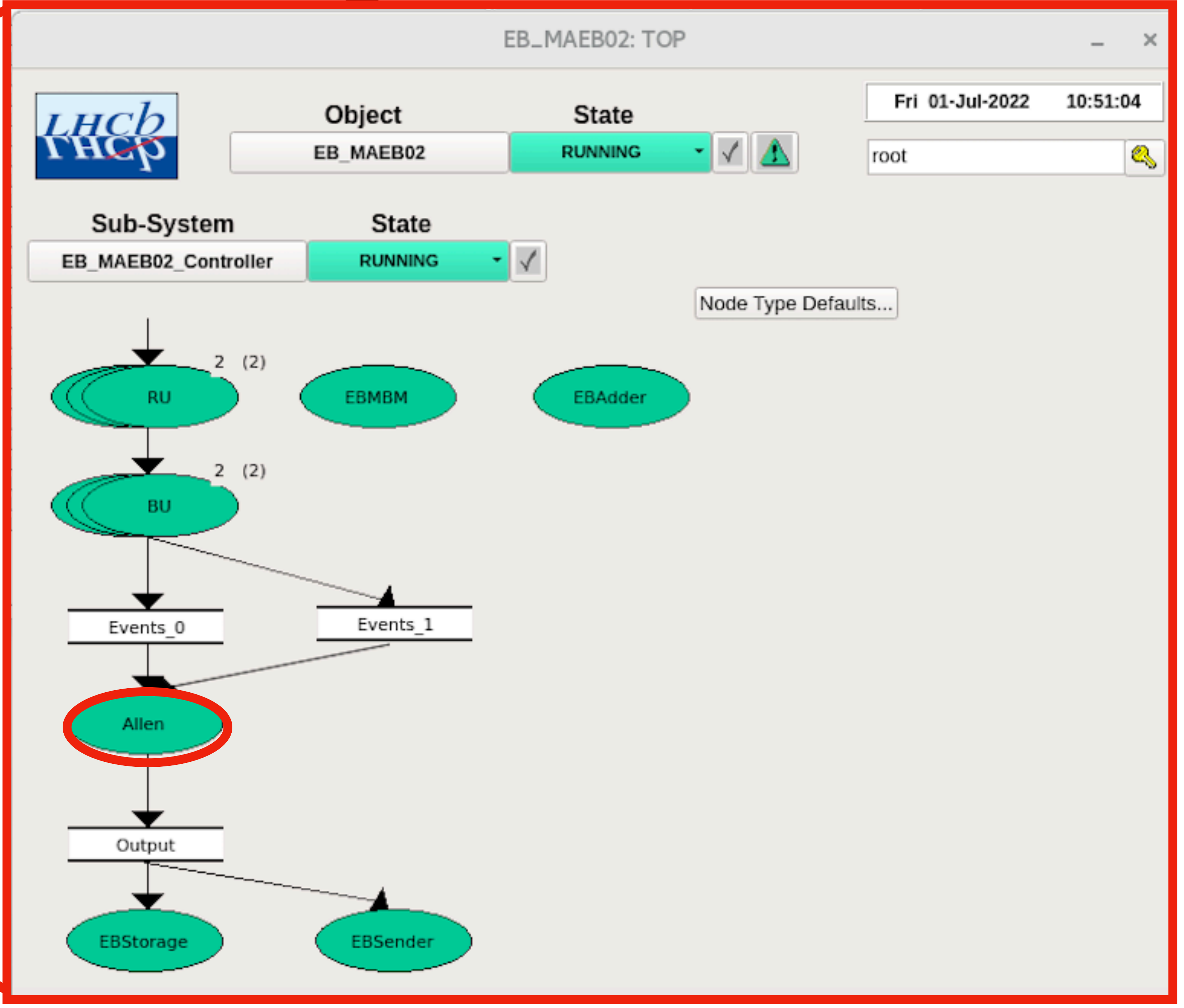
Data taking



- LHCb has been exercising its DAQ in parallel to the LHC commissioning
- Sub-set of detectors (Calorimeters, Muon stations, PLUME) already in the global partition of the Experiment Control System (ECS)
- System running 24/7 in parallel to sub-detector commissioning activities

Data taking

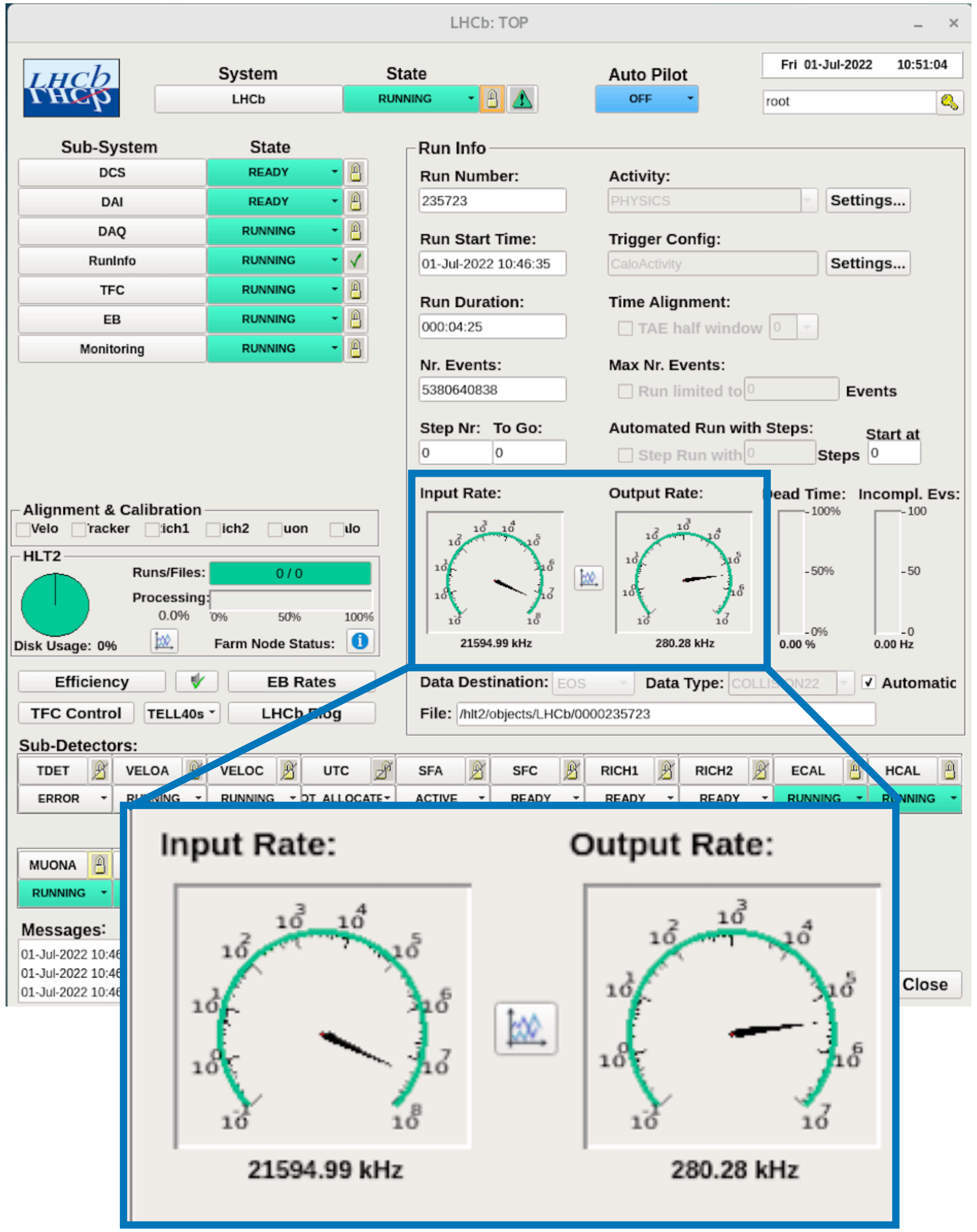


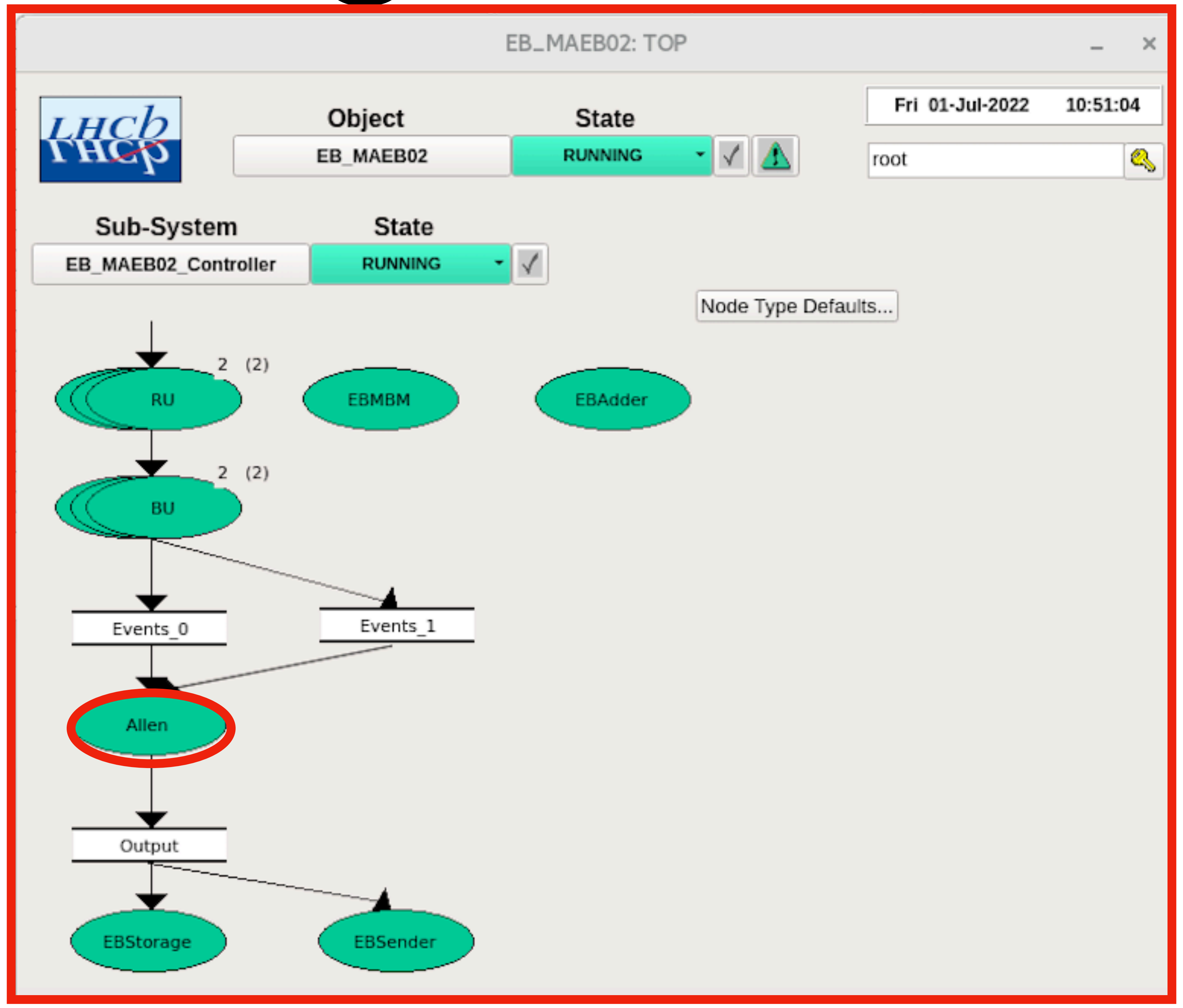


- ~200 GPUs are installed in the EB
- HLT1 is already included in the global partition

Daniel Cámpora

Data taking





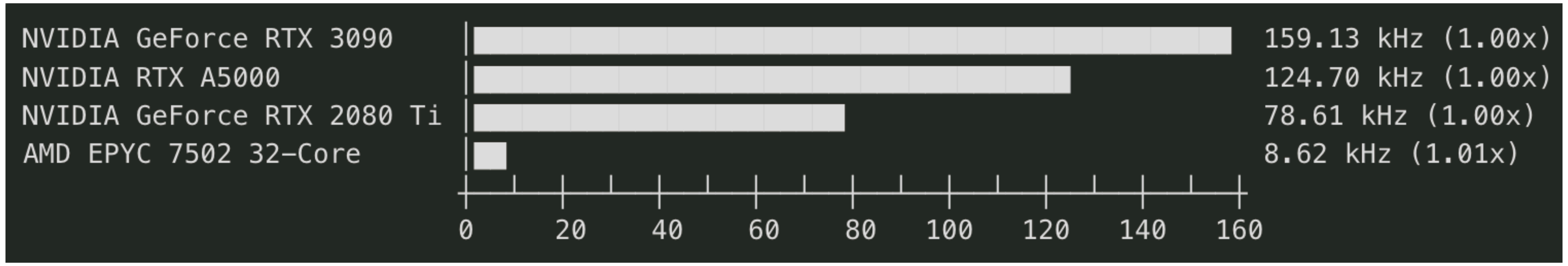
- ~200 GPUs are installed in the EB
- HLT1 is already included in the global partition
- Here shown: Triggering on calorimeter clusters @ 20 MHz

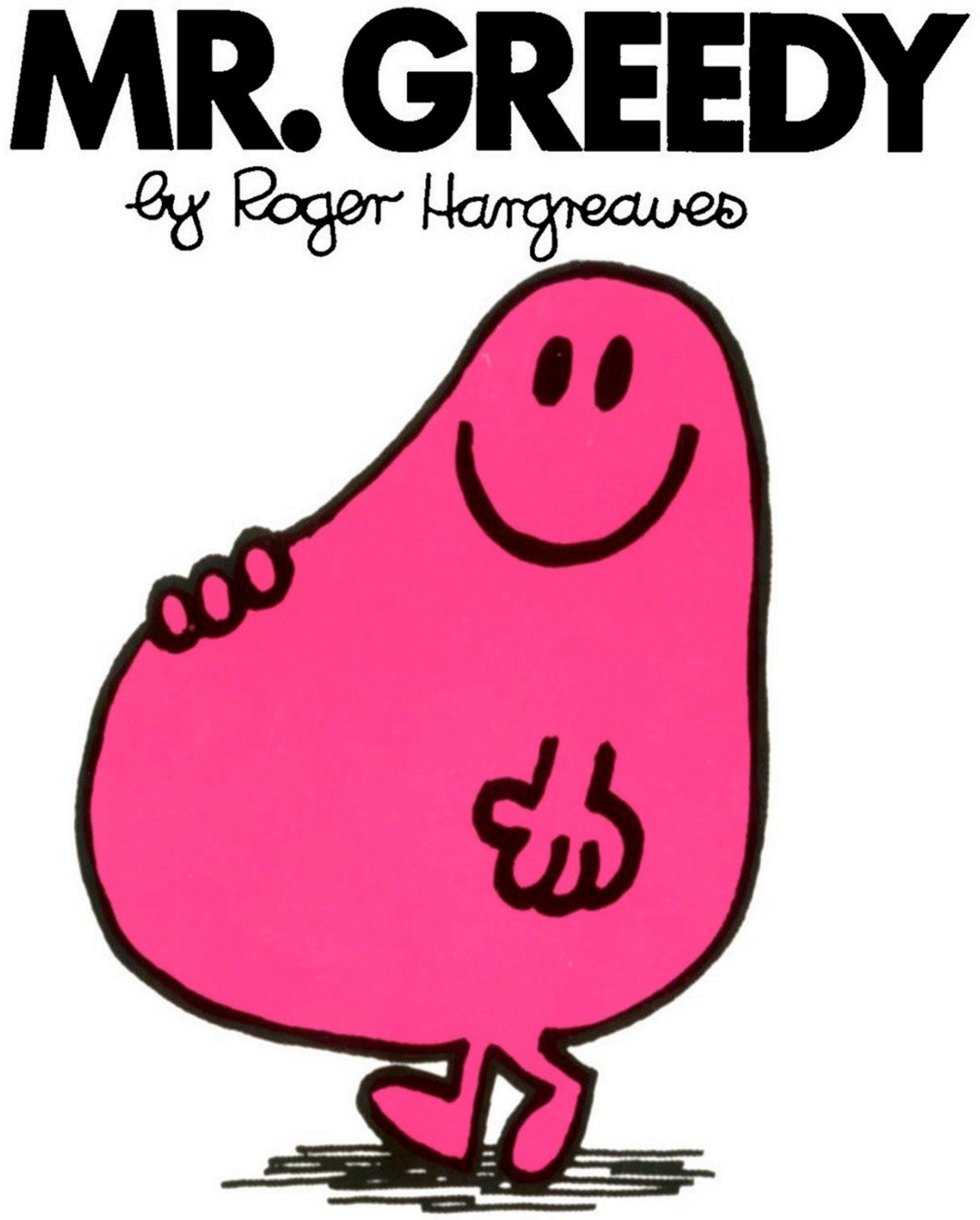
Greedy because we can

- For the moment, we are running either of

  `hlt1_pp_matching_no_gec`

  `hlt1_pp_no_gec_no_ut`





- *No UT* means drop in physics efficiency, but not in throughput.
- *No GEC* costs about 20% in throughput.
- A5000 is running at 125 kHz for `hlt1_pp_no_gec_no_ut`, so one expects drops if we run at full rate and luminosity

So are we good?

- Rate of current runs is about 20 MHz, held back by various issues
  - Recent retina update, now it's RICH...

- In terms of pileup, this year we will run at mu=1. Next year, mu=7.
  - Nu, mu and pileup definitions: https://twiki.cern.ch/twiki/bin/view/LHCb/NuMuPileUp
  - Even when we run at 30 MHz, Allen will handle it this year.

- For the upcoming period we will need to do something to keep using the current sequences and scale up.

Milestones - First trigger on real data

- Test the High Level Trigger 1 (HLT1):
  - Copy data to / from GPUs, perform decoding and reconstruction algorithms, select based on physics cuts
  - For now only based on calorimeter data: Decode 3x3 ecal clusters, trigger on > 400 MeV
  - What actually happened:

Saturday 28/05:

- Since 9:12: `CaloActivity`, triggering on ADC, no decoding fix
- 12:55: Calo experts suggest to switch to cluster trigger line, 400 to 2000 MeV
- **14:26:** Decoding fix implemented, first proper trigger from this point on. **Beam dump at 14:30.**

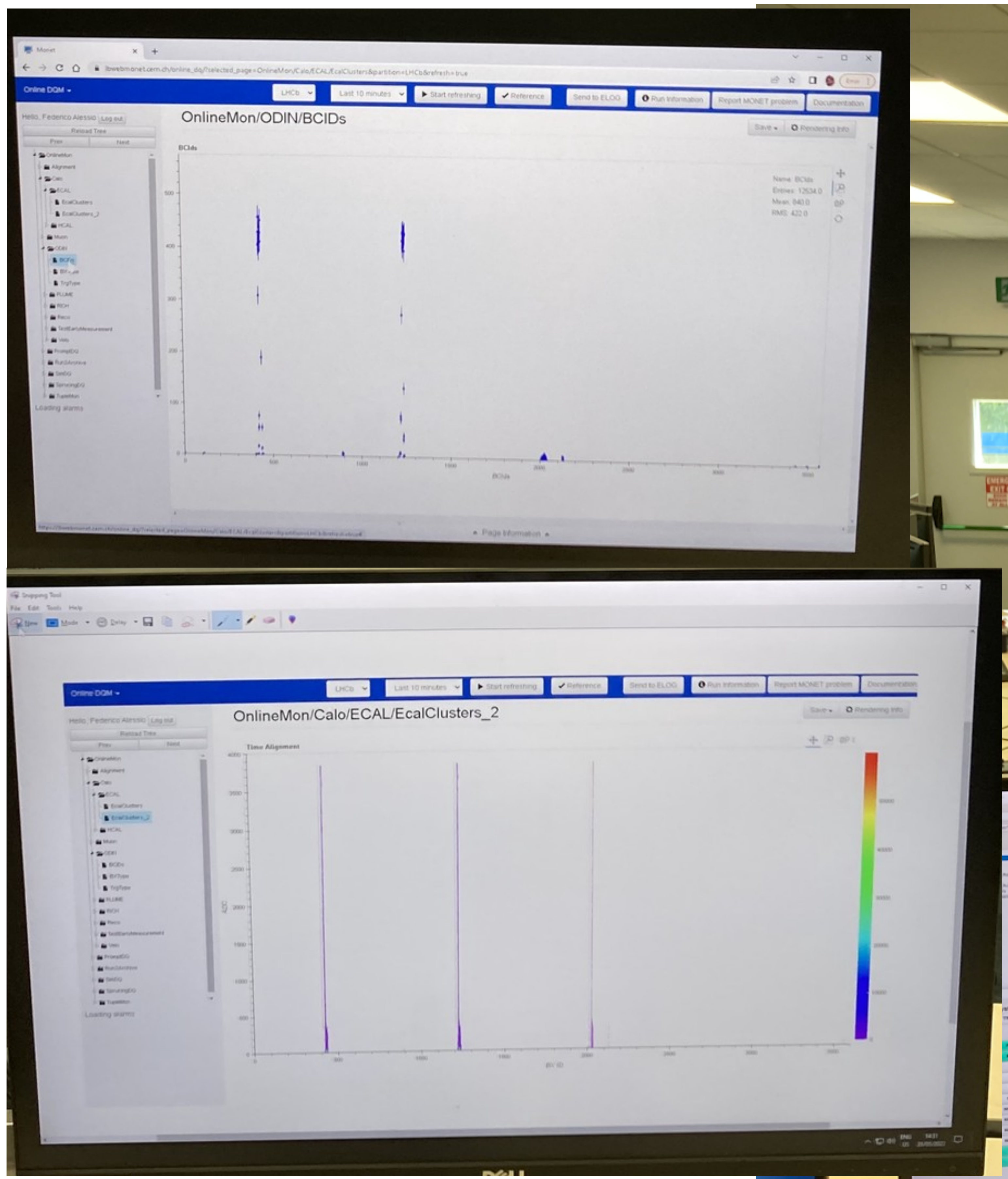
Sunday 29/05:

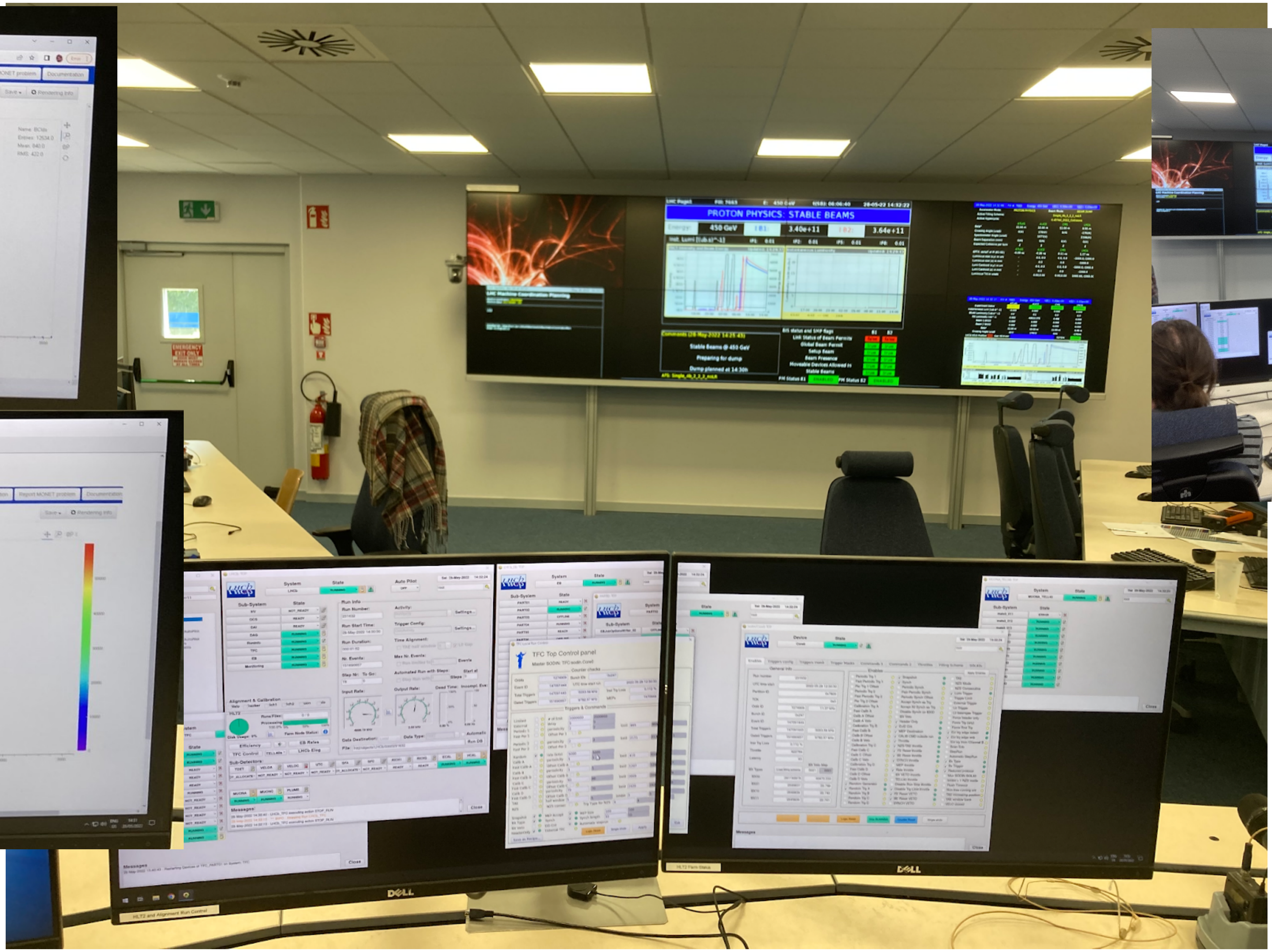
- HLT1 ran all day with no major hiccups

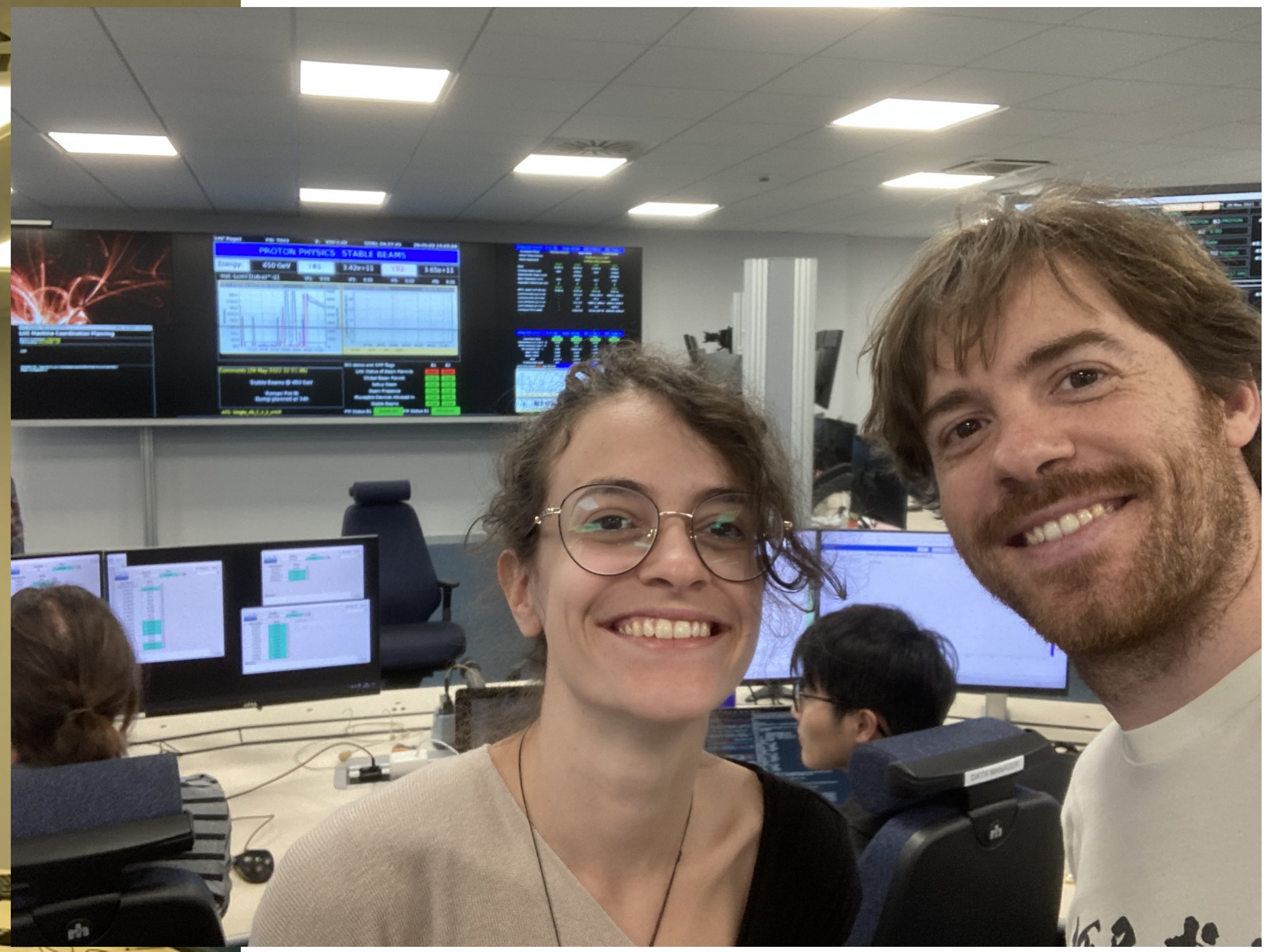
Daniel Cámpora

Milestones - First trigger on real data

- First trigger on real-data in LHCb Run 3 (28/05/22)

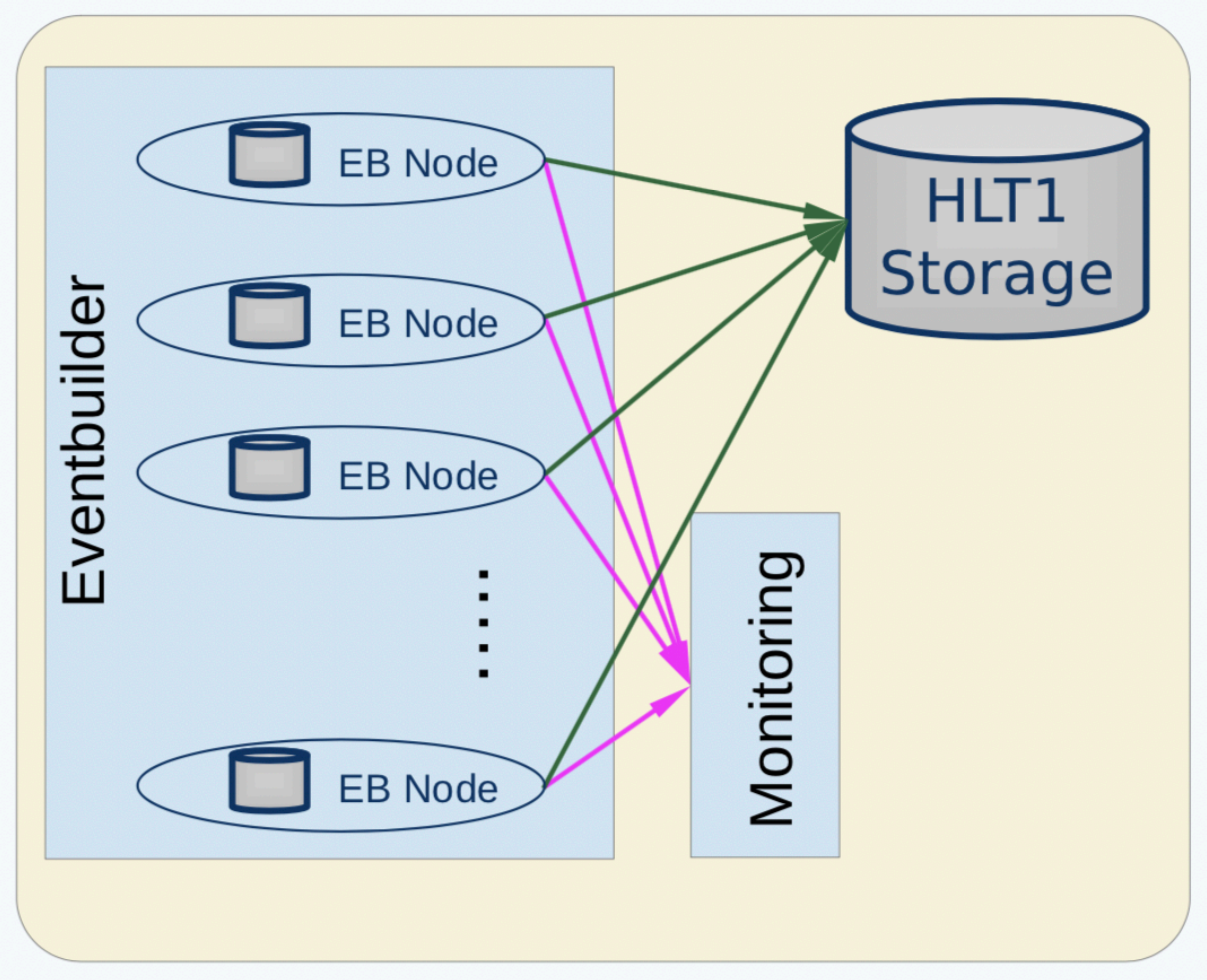






Milestones - DAQ

- Large-scale Data Acquisition test



Data-flow:

- Up to 40 Tbps of data from the subdetectors if received by FPGA cards (PCIe40) hosted in the Event Building (EB) servers
- The EB network protocol bring the subdetector data fragments produced in a single event together and then group these events together into Multi-Event Packets (MEPs)
- Allen receives MEPs from the EB and processes them, producing output files in Markus' Data-Format (MDF)

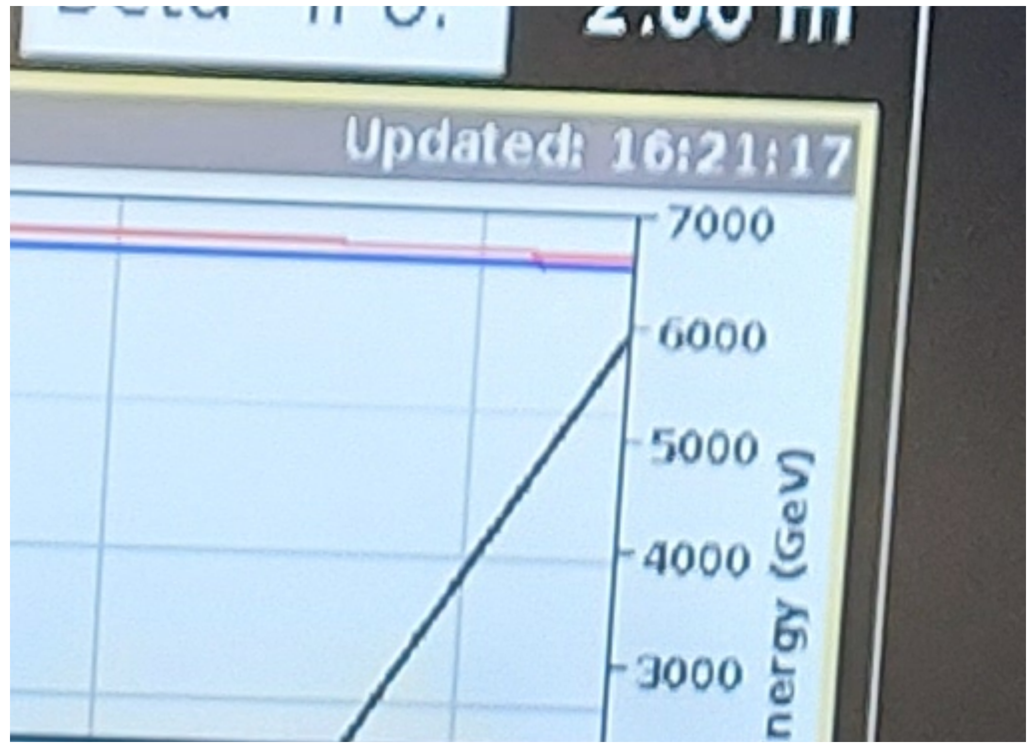
Milestones - DAQ

2/06/22, 10 PM: First successful test at 10 MHz

- Take all TELL40s from all subsystems and try running the full DAQ system and push the limits of the TELL40, EB and storage
- Tested 1-2-5-10-15 MHz with no major issues encountered
- Latest tests pushing the rates to 20-30 MHz achieved in the next days after solving an issue

Milestones - Run at design energy

- Highest energy in a particle physics accelerator (13.6 TeV, 05/07/22)

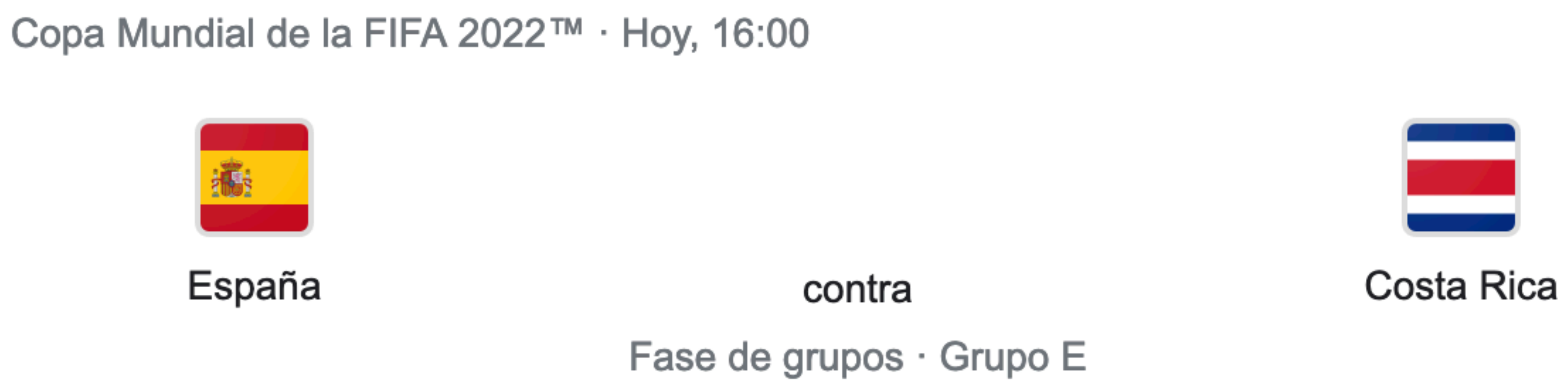


Daniel Cámpora

Conclusions

- The R&D of many years (at least since 2012!) has finally been realized
  - LHCb has a new detector, a new DAQ and a new trigger
  - The specification has changed quite a lot, only possible with a dedicated collaboration and lots of hard work (it's also fun)
- Software has changed along the way, very different from 10 years ago
  - New architectures available, exciting time to develop new software
- Commissioning is a rocky road, but extremely rewarding
  - We are preparing the physics of tomorrow
  - You can make a great impact!

Future work


Copa Mundial de la FIFA 2022™ · Hoy, 16:00
España
contra
Costa Rica
Fase de grupos · Grupo E

Thank you for your attention!