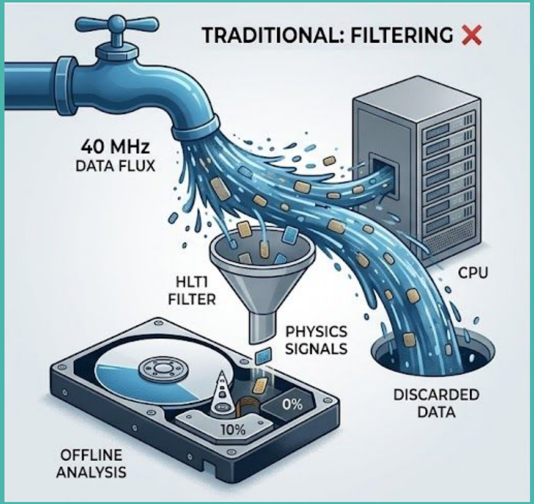
Real Time Analysis

Carla Marin
cmarin@fqa.ub.edu

GA: 101077940
European Research Council
Established by the European Commission
Universitat de Barcelona
Ministerio de Ciencia e Innovación
Agencia Estatal de Investigación

The Data Flood and the Trigger Challenge



Data rates at HEP experiments

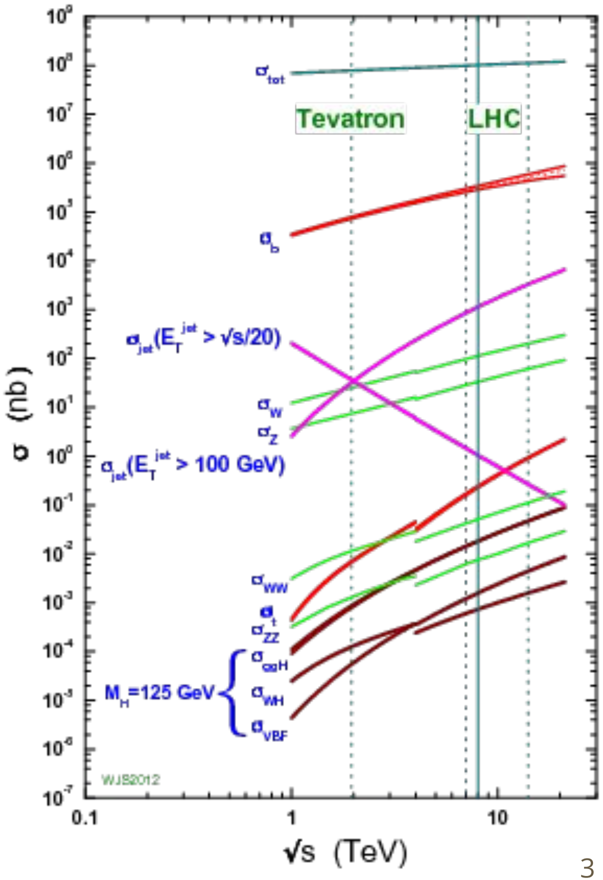
events/s = σ (cm^2) x Luminosity (cm^{-2} s^{-1})

LHC: pp collisions at 40 MHz with L ~ O($10^{33\text{-}34}$) cm^{-2}s^{-1}

- O(10) Z/s, W/s
- O(10^5) bb/s
- O(10^8) events/s

Experiments: O(10^7) readout channels

LHC experiments produce O(1–10^2) TB/s

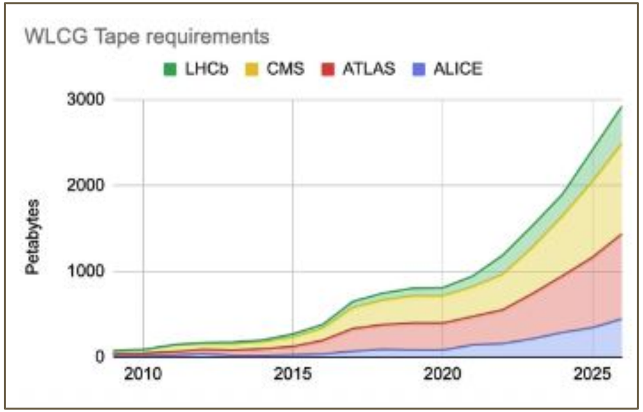


Technical limitations

Ideally: store all information produced at any experiment

Reality:

- writing data to disk takes time
- elevated disk/tape cost
- LHC numbers: 10^2 TB/s x 10^6 s/year = **10^5 PB/year**!



[G. Merino - Distributed computing]

→ can write only a fraction of the information

Trigger systems

Define **conditions** under which an event shall be **recorded**

- select interesting physics events = Signal
- reject non-interesting events = Background
- **Goal**: reduce recorded data size while enabling physics programme

Trigger systems

Define **conditions** under which an event shall be **recorded**

- select interesting physics events = Signal
- reject non-interesting events = Background
- **Goal**: reduce recorded data size while enabling physics programme

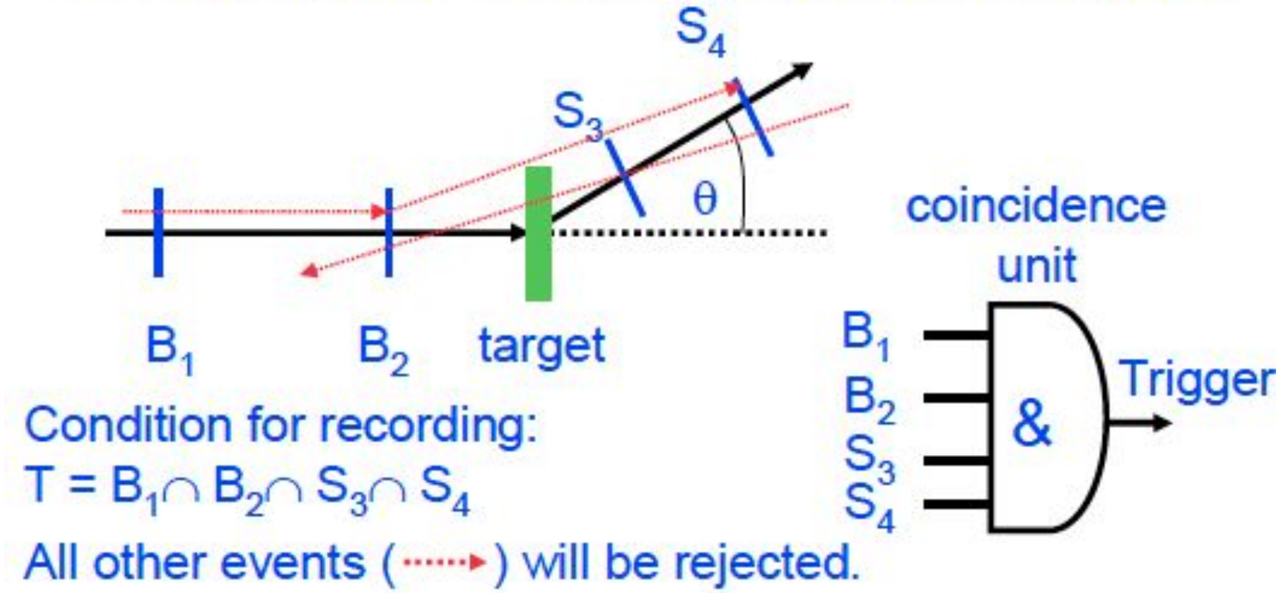
Taking the "decision" and recording data takes time (τ_{rec} ~ ms/evt)

- ideally: rate R [evts/s] < $1/\tau_{rec}$
- otherwise → **Deadtime** = loss of events
  - Eg LHC: R = 40 MHz → τ_{rec} < 25 ns

$\boxed{\text{Throughput = evts processed/s} = 1/\tau_{rec}}$

Trigger example

Scattering experiment, want to select only particles deviated angle θ



Trigger levels

HEP experiments much more **complex**:

- several **tracking** stations → pattern recognition + track fitting
- large **calorimeters** → clustering, energy & position corrections
- complex **particle identification** → match info from various subsystems

Trigger levels

HEP experiments much more complex:

- several tracking stations → pattern recognition + track fitting
- large calorimeters → clustering, energy & position corrections
- complex particle identification → match info from various subsystems

Divide & conquer: trigger levels of increasing complexity & selectivity

1. Coarse info from all events → fast hardware level (L0/L1)
2. Refined info from selected events only → software level (HLT)

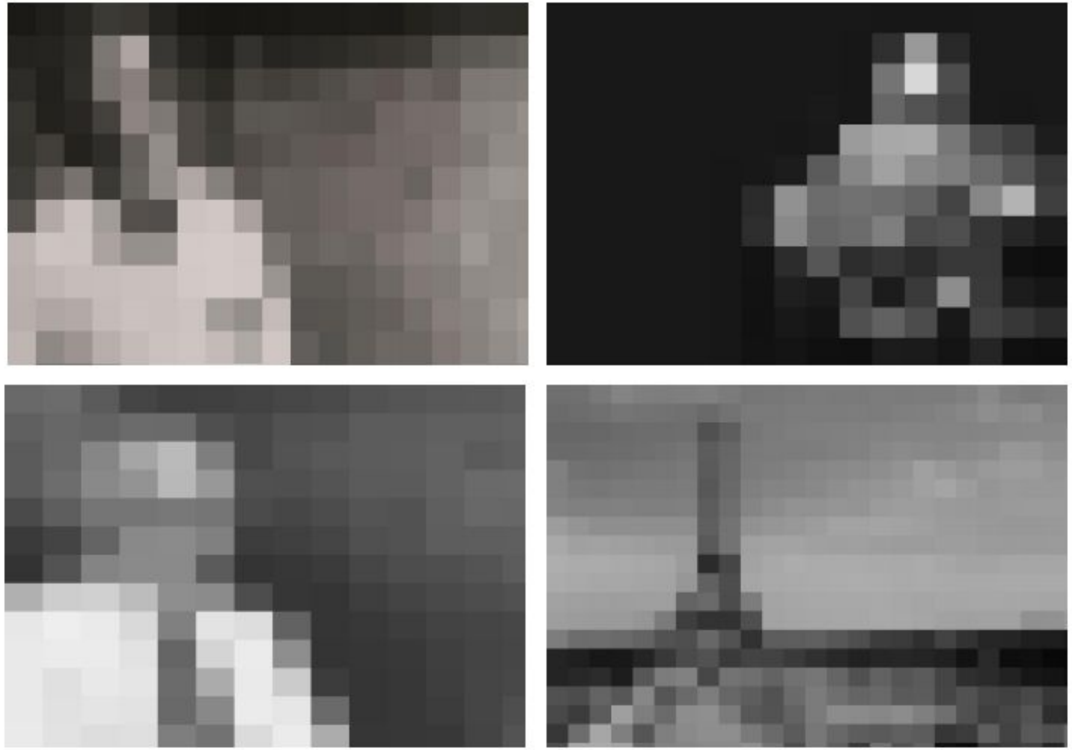
Data from previous level temporarily stored (buffer) until decision is taken

Trigger levels example: Higgs

Higgs Selection using the Trigger

Level 1:

Not all information available, coarse granularity

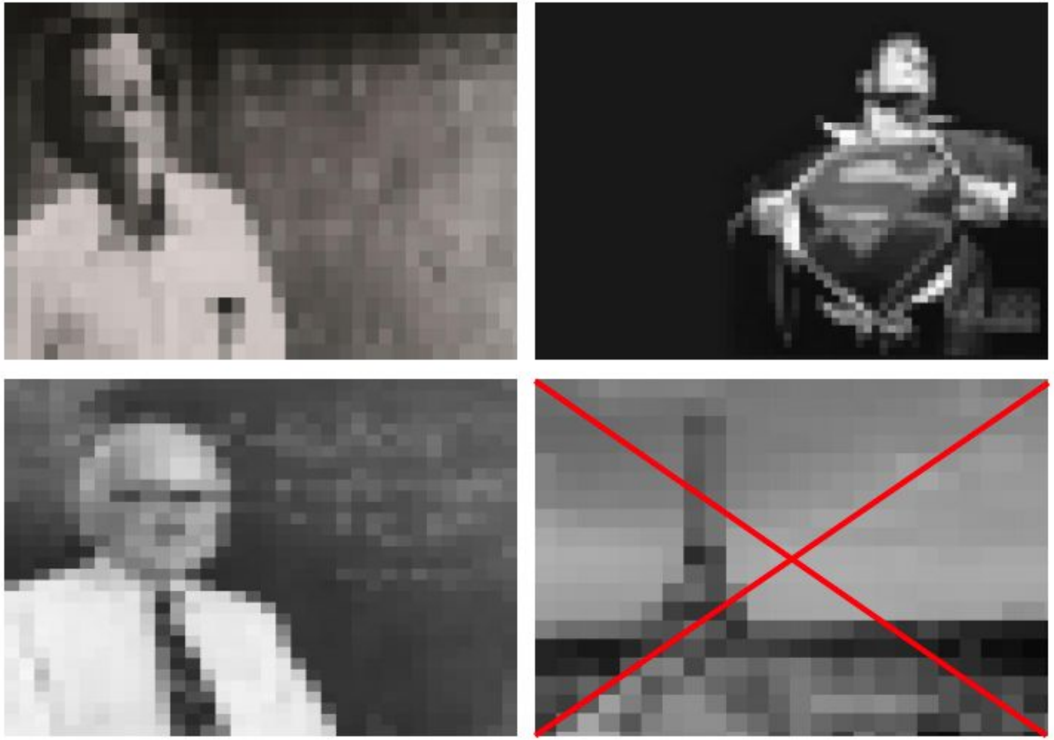


Trigger levels example: Higgs

Higgs Selection
using the Trigger

Level 2:

Improved reconstruction
techniques, improved
ability to reject events

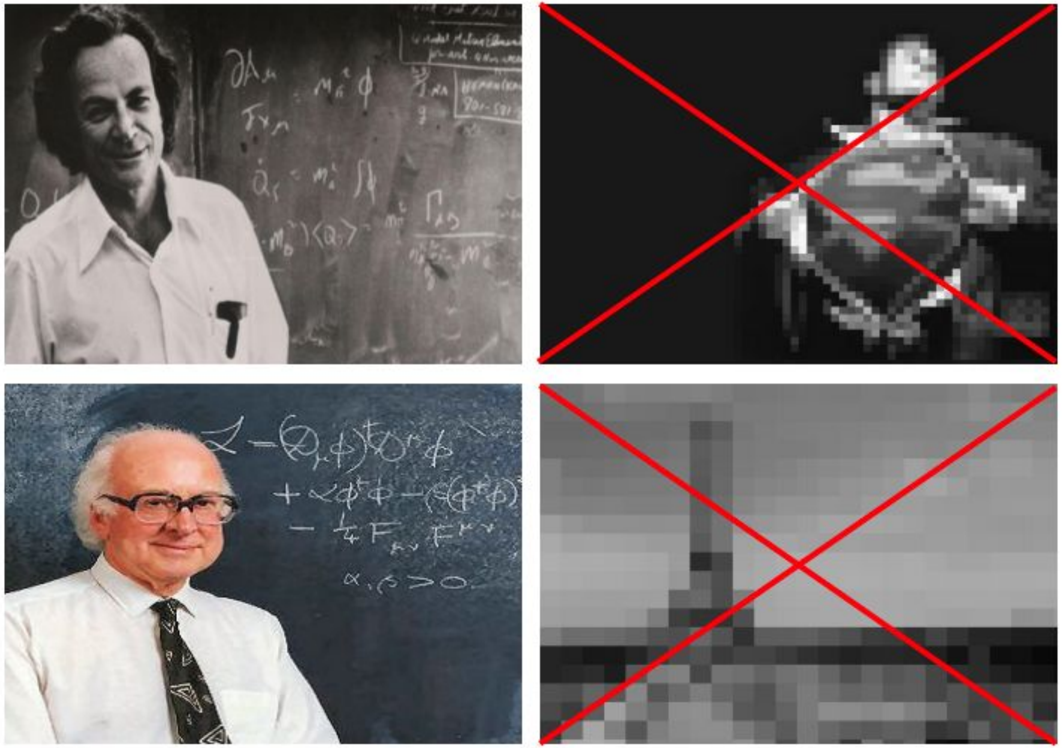


Trigger levels example: Higgs

Higgs Selection
using the Trigger

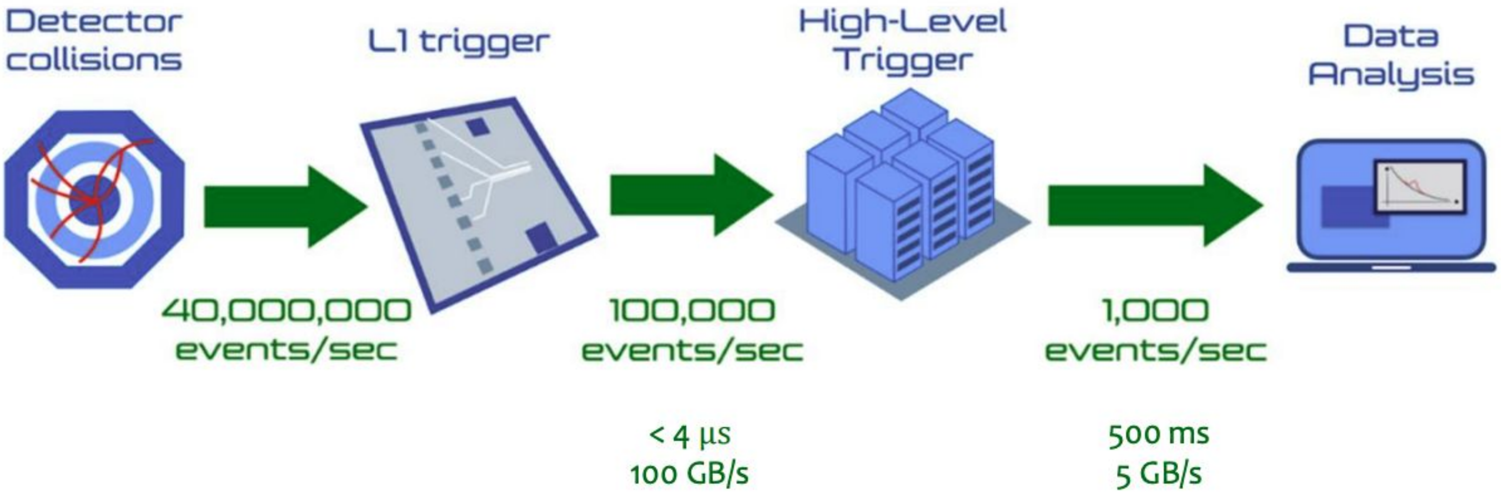
Level 3:

High quality
reconstruction
algorithms using
information from
all detectors



Classic trigger scheme

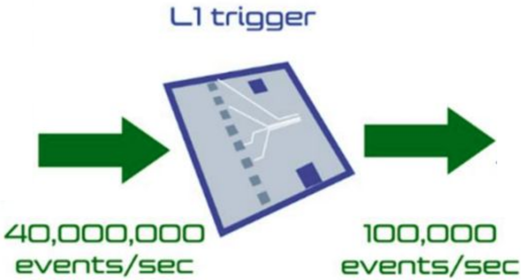
Eg: CMS trigger in Run 3



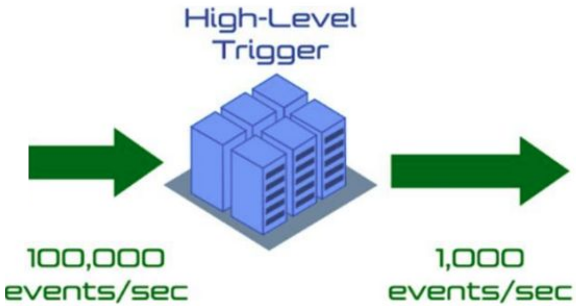
[S. Folgueras @COMCHA workshop]

Classic trigger scheme

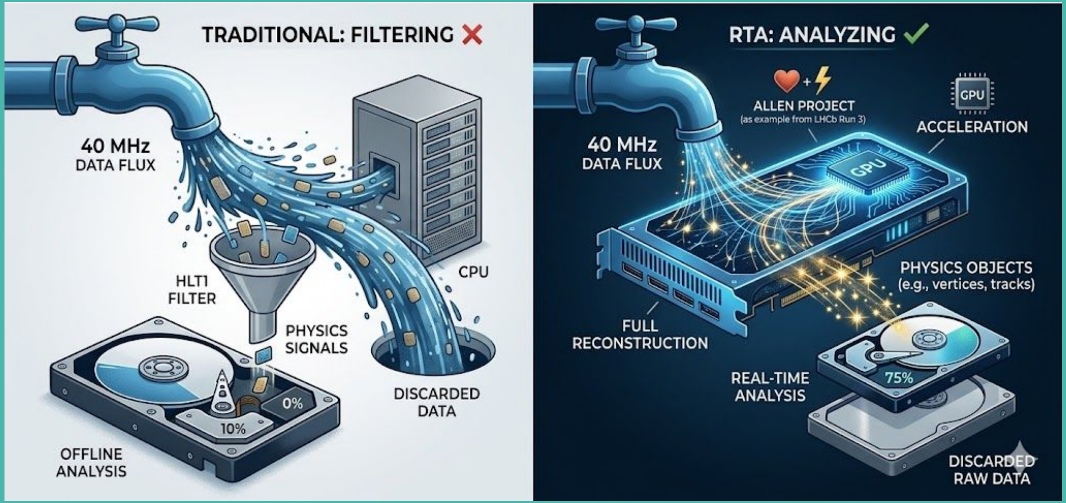
- **Level 1** (L1): based on hardware
  - very fast decision, O(μs)
  - based on basic information (eg Energy from calo/muons)
  - selects 1/100 - 1/10 of input events and writes to disk buffer
  - runs on custom electronic boards



- **High Level Trigger** (HLT): based on software
  - fast decision but less constrained than L1
  - more complex info from other/all subdetectors
  - typically runs on PC farm (O(1k) nodes)
  - can include sub-levels:
    - HLT1: partial reconstruction
    - HLT2: full reconstruction



The Paradigm Shift:
from Triggering to Real Time Analysis

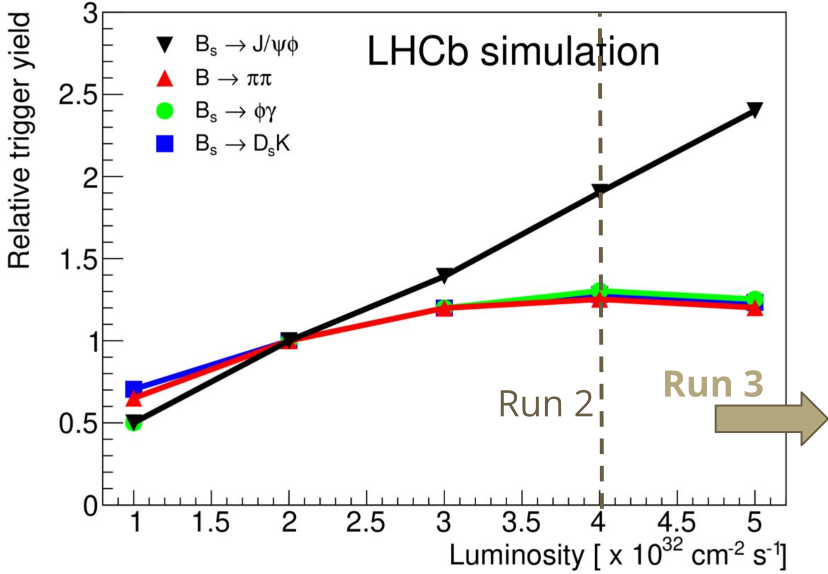


JINST 19 (2024) P05065

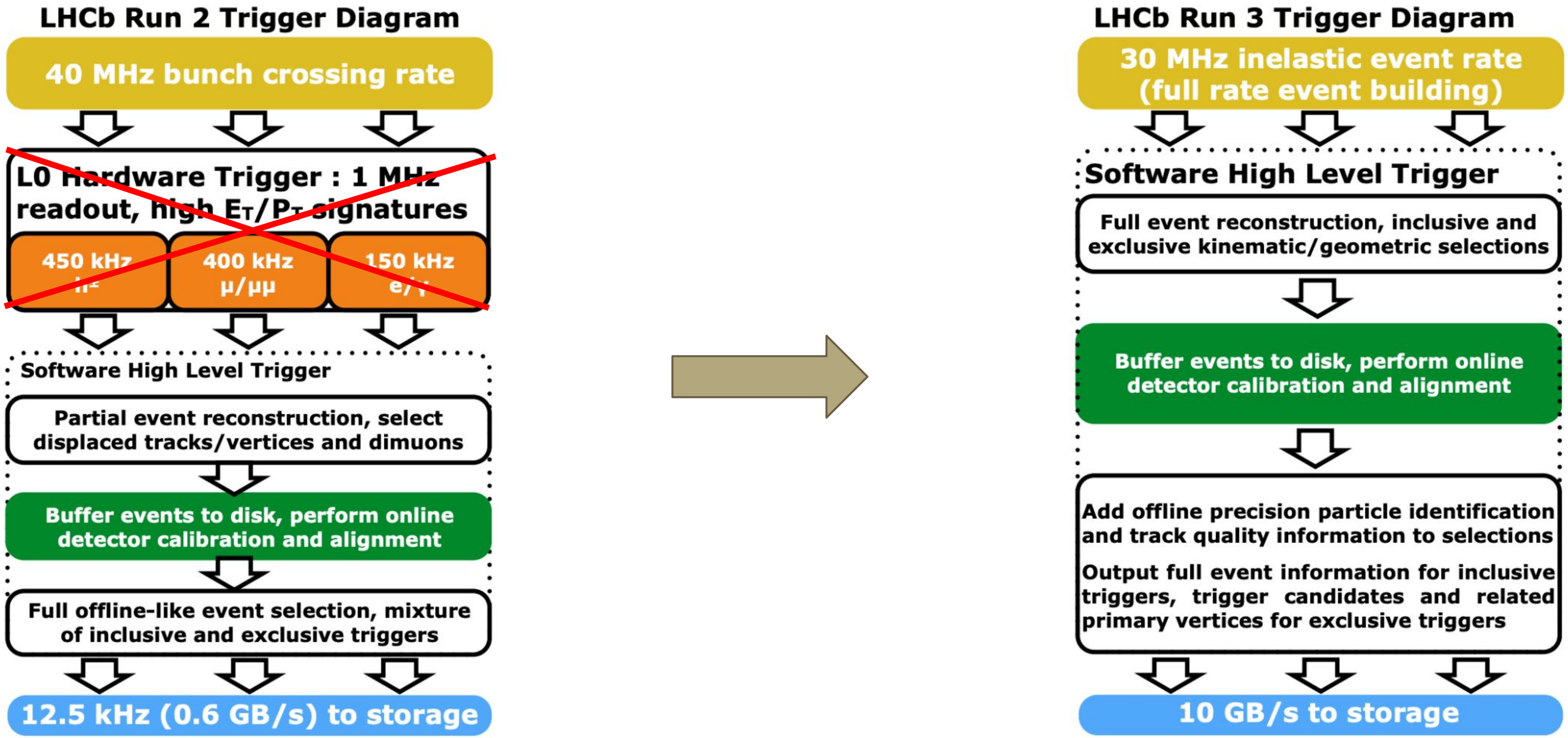
The classical trigger limitation

Classical trigger: hardware + software stages

- Limitation: HW trigger rate limit
  - 100 kHz - 1 MHz
  - requires tight cuts on calorimeter energy
- Low momentum signals saturate the HW trigger rate
  - c & b-physics signatures
  - low momentum resonance searches



Trigger-less readout @LHCb Run 3

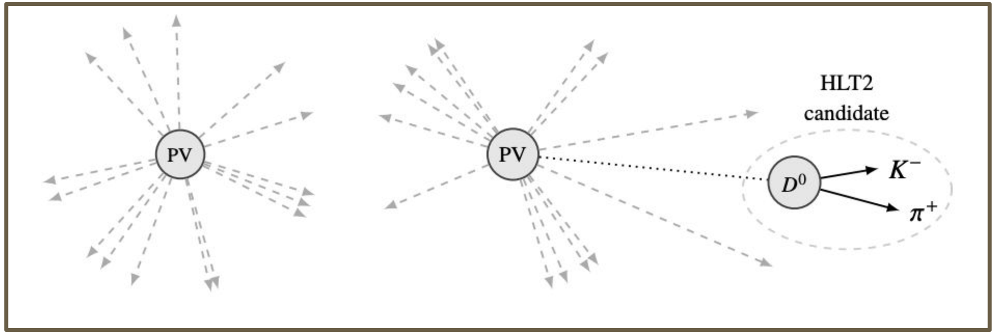


*also ALICE uses continous triggerless readout in Run 3

Filtering vs Analysing

Classic Trigger:

- select few signal events (O(%))
- keep all their information



Real Time Analysis:

- save many events
- keep only the relevant information → Data reduction
- requires detailed analysis of all events to identify relevant events & info

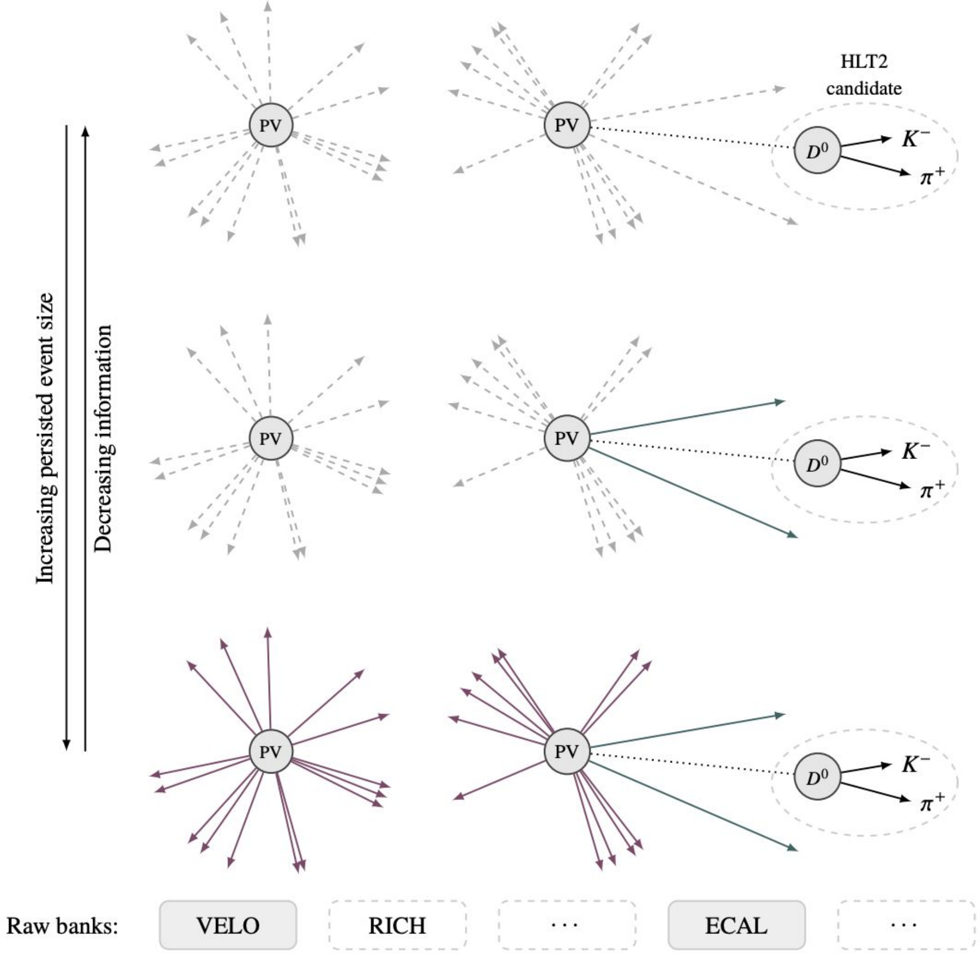
JINST 14 P04006

LHCb: turbo model

Flexible persistence model:

- **Turbo**: signal only
- **Selective**: signal + selection of reconstructed objects
- **Complete**: all reconstructed objects
- **Raw event**: detector hits

Persistence method	Average event size (kB)
Turbo	7
Selective persistence	16
Complete persistence	48
Raw event	69

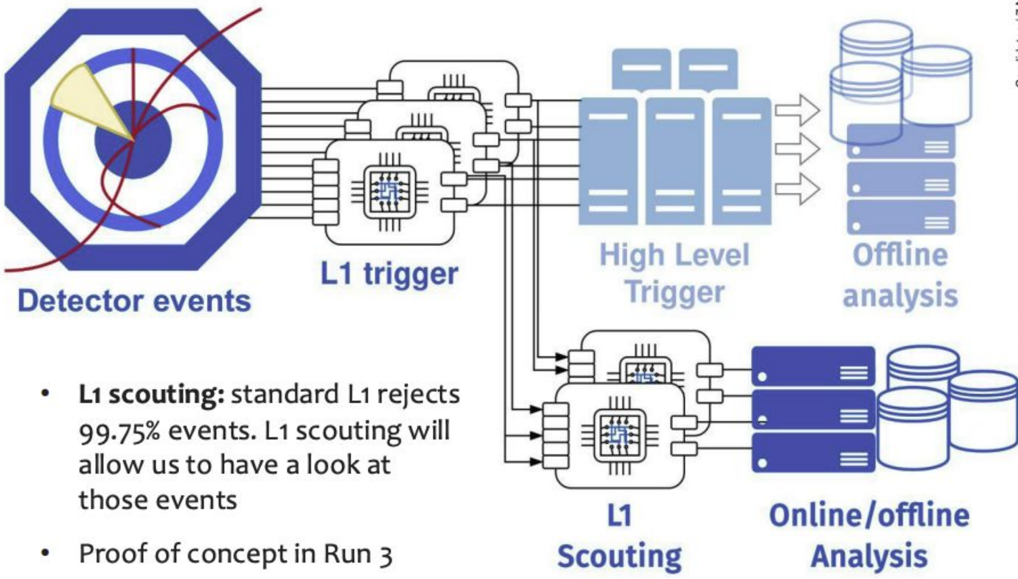


CMS and Atlas: Data Scouting

Store HLT(L1) objects (x100 smaller)

- select **low energy** candidates that wouldn't pass the trigger
  - eg: very low-mass searches
- **store HLT(L1) object** directly
  - O(10 kB/evt) vs O(1 MB/evt)
- offline analysis on HLT(L1) objects
  - eg [Observation η → 4μ]

→ can select many more events BUT at lower precision (L1/HLT quality)

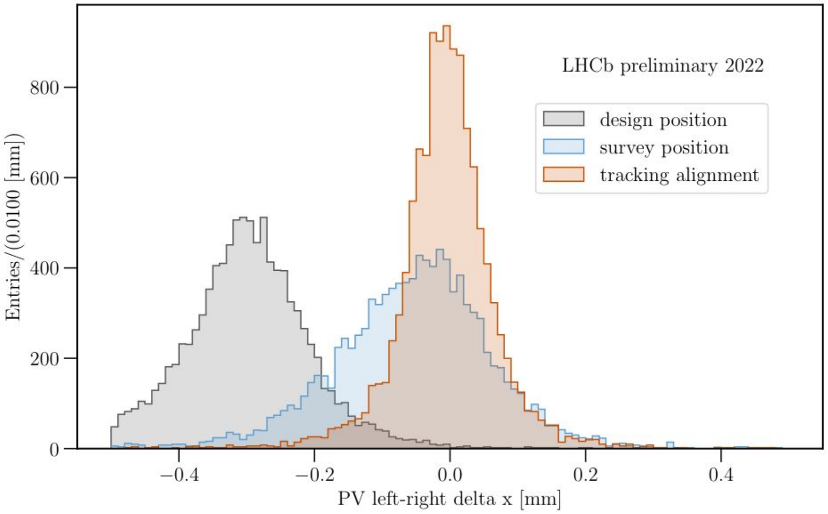


[S. Folgueras @COMCHA workshop]

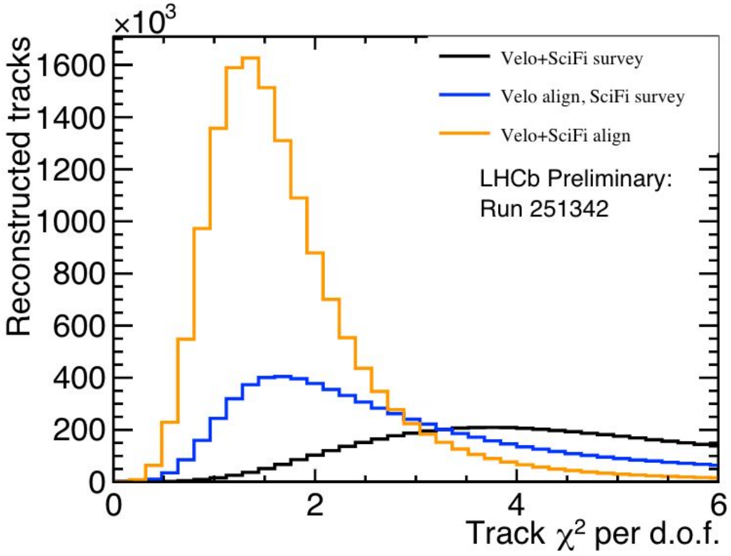
Enabling Technologies

Online Alignment & Calibration

Detector alignment & calibration is crucial for high quality reconstruction



LHCB-FIGURE-2022-016



LHCB-FIGURE-2022-018

Online Alignment & Calibration

Detector alignment & calibration is crucial for high quality reconstruction

Classical trigger:

- **offline** alignment & calibration (weeks/months later) on triggered data
- full-quality reconstruction runs offline using offline calibration

Real Time Analysis:

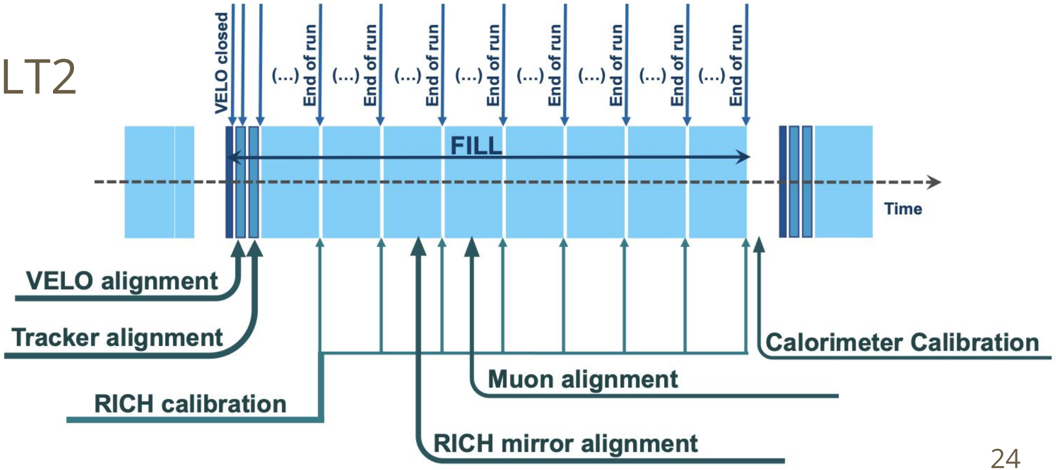
- calibration runs **online** (in seconds/minutes) before final trigger level
- objects reconstructed by HLT have full quality → ready for analysis!

Online Alignment & calibration @LHCb Run 3

Use HLT1 selected data to derive alignment and calibration constants

- dedicated task for each detector, running on HLT2 farm
- jobs run automatically when enough events collected
- results stored in conditions database

⇒ enables offline-level quality in HLT2



Heterogeneous Computing: why?

Computing needs in a triggerless readout:

- **Throughput**: process 10^7 events/s in software
- **Complexity**: O(100-1000) tracks per event, massively parallel data

Traditional **CPUs**:

- excellent for sequential tasks but harder to exploit for parallel tasks
- CPU power is not increasing at same speed anymore
- high energy consumption

Heterogeneous Computing: what?

FPGA: Field Programmable Gate Array

- Extreme speed: can reach μs latency
- Configured for specific task (logic gates)
- Massive I/O of bits/hits in simple algorithms

Custom boards placed on-detector

eg: Atlas & CMS L1 triggers

GPU: Graphic Processing Unit

- Ideal for parallel computing: single instruction, multiple data (SIMD)
  - eg: curvature calculation for 1000 tracks
- Easy to implement complex algorithms (vertexing, tracking, etc.) in C++/CUDA

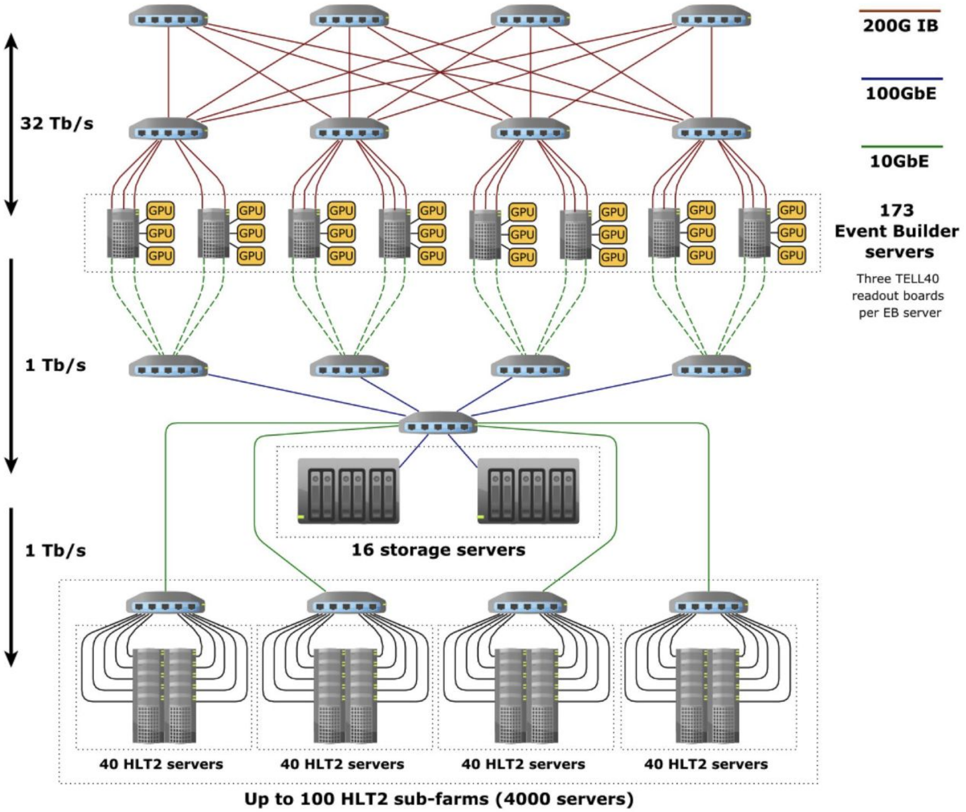
Placed in server farms together with CPUs

eg: Allen@LHCb HLT1 on GPUs at full LHC rate

Heterogeneous Computing @LHCb Run 3

Hybrid architecture:

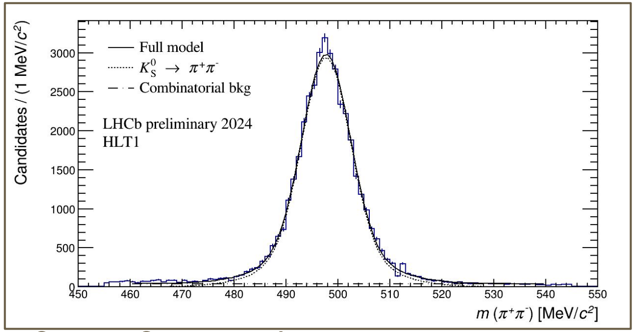
- DAQ: **FPGAs** in detector electronics
- HLT1: **GPUs** installed in Event Builder servers
- HLT2: **CPUs** in Event Filter Farm



Risks and Validation of Real Time Analysis

Fear of Missing Out

- **Permanent decisions**: unsaved info lost forever
  - discard raw event (Turbo) → cannot reprocess with full info offline
  - selection bug discovered after data taking → data lost
  - imperfect online alignment/calibration → affects offline analysis quality



- **Mitigations**:
  - online comprehensive **monitoring** of standard candles
  - **unbiased stream** for selection validation
  - **full stream**: 25% events selected inclusively → re-analysis of new future ideas
  - **calibration stream**: 5% events selected with full detector information → continuous calibration improvement for future runs and impact of imperfections on analyses

Software crashes

Full software readout implies software cannot crash at any time!

- software crash = data lost!

Mitigations:

- robust design against unexpected features/data
- continuous integration and testing of software: GitLab + nightly builds
- massive testing on simulation and detector generated data (w/o beam)
- always back-up stack ready for production

Systematic Uncertainties

Online calibration/reconstruction imperfections can limit precision of offline analysis

Mitigations:

- ultra-precise validation in simulation and first data
- **RecoMon**: online monitoring of reco quantities
- **CalibMon**: online monitoring of calibration metrics

Real Time Analysis in Astroparticles
From Data Storage to Immediate Reaction

Multi-messenger alerts and Transients

Goal: catch transient phenomena before they fade

- CTA **30 seconds challenge**: Gamma-Ray Bursts (GRBs) fade in sec/mins
    - **RTA:** Real-time reconstruction of air showers to generate "Science Alerts"
    - **Impact:** allows CTA to pivot and other observatories (optical/X-ray) to follow up within discovery window

- SN **Early Warning System** (SNEWS): ν escape super-nova before photons
    - RTA: real-time monitoring of ν flux across experiments
    - Alert astro community to observe first light of stellar collapse

Background rejection

Goal: find ultra-rare signals in within huge backgrounds

Continuous Streaming (IceCube / KM3NeT)**:**

- **Challenge:** huge background (eg ^{40}K radioactivity or atmospheric μ) and very rare neutrino interactions
- **RTA:** "Online Triggering" via spatial and temporal coincidence. FPGAs with advanced ML filter out uncorrelated noise at ns level
    a. smart object identification keeps sensitivity to low-energy events
- **Result:** huge data reduction, keeping only potential physics-rich tracks

Conclusions and High-Luminosity Future

Summary

- **Paradigm shift**: from triggering to Real-Time Analysis
    - Triggerless readout: LHCb and Alice Run 3
    - Data reduction vs event selection: scouting vs selective persistence
- **Technology** is crucial to achieve this
    - CPU clock speed has plateaued
    - Heterogeneous computing: FPGAs and GPUs
    - Software portability is crucial to keep flexibility
- The drawback: no safety net → testing & validation more crucial than ever
- RTA is a shared technology between HEP and Astroparticles

Conclusions

Real-Time Analysis (RTA) bridges detector to physics publication, it’s not just a “trigger” but the analysis itself

HL-LHC challenge: x10 higher pileup will require ML driven triggers in FPGAs and GPUs (or new tech.)

RTA shifts limit of discovery: “how much we can **store**” → “how much we can **process**”

- every collision is a potential measurement rather than a discarded byte

Thanks for the attention

BACK-UP

Trigger vs RTA summary

Feature	**Classical Trigger**	**Real-Time Analysis (RTA)**
Primary Decision	Binary (Keep or Discard)	Processing (What information to keep?)
Data Format	Full Event (Raw Data)	Reconstructed Objects (Analysis Ready)
Main Constraint	Storage Bandwidth (Output MB/s)	Computing Power (CPU/GPU cycles)
Calibration	Offline (Weeks/Months later)	Online (Real-time or quasi-online)
Physics Reach	High-energy / High-PT	Low-energy / High-statistics
Workflow	Sequential (Online -> Offline)	Integrated (Online = Analysis)
Storage Strategy	Small fraction of events, full detail	Large fraction of events, reduced detail

HLT1 performance

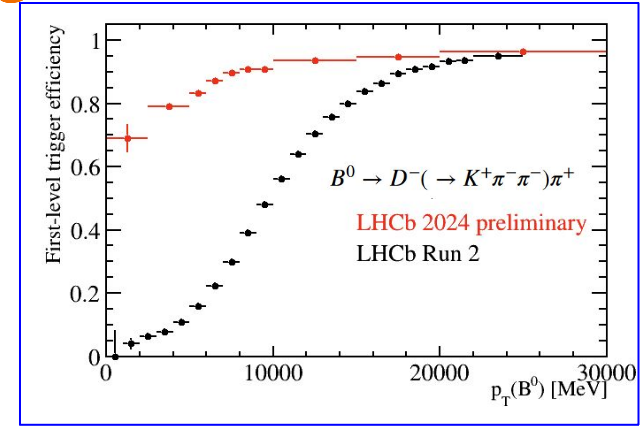
LHCb-FIGURE-2024-006
LHCb-FIGURE-2024-007
LHCb-FIGURE-2024-014

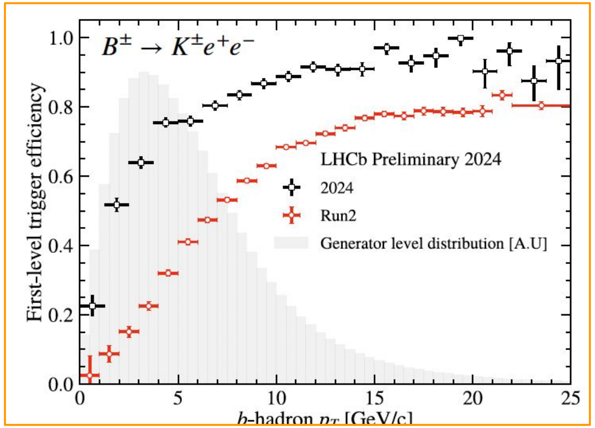
Successfully processing 30 MHz at pile-up of 5.3.

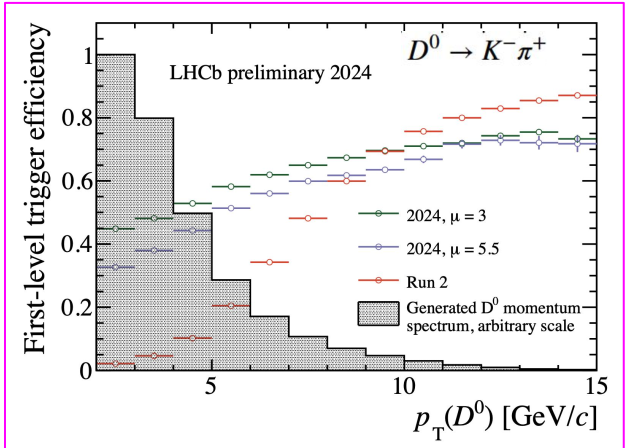
Large gains in efficiency compared to Run 2 for:

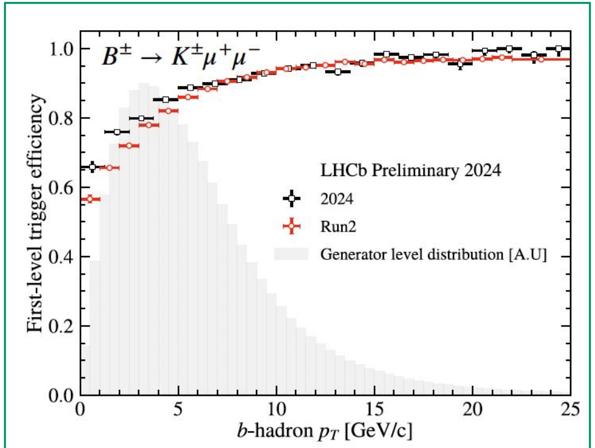
- hadronic modes
- electron modes
- charm decays

Even a bit for muon modes





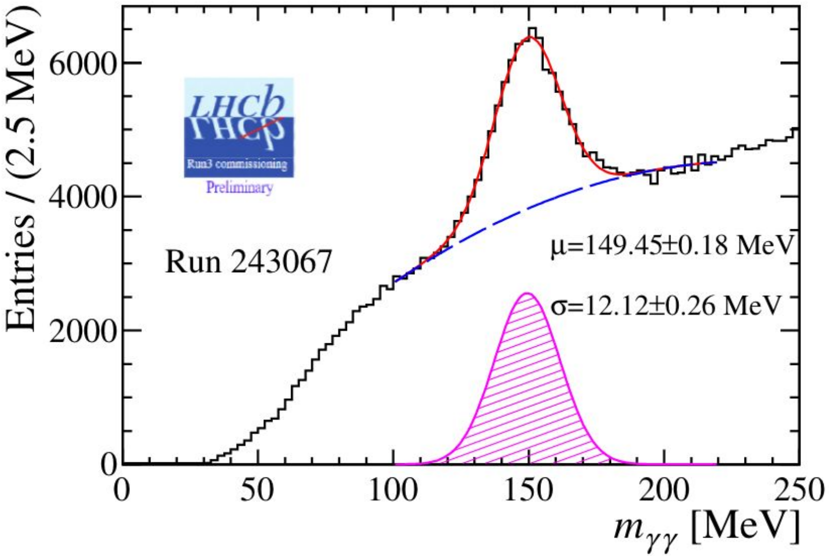




Calibration

[LHCB-FIGURE-2022-019]

ECAL calibration from $\pi^0 \to \gamma\gamma$ events



Calibration →