Allen: GPU-based event processing framework at LHCb

Da Yu Tou (杜大佑)
on behalf of the LHCb Experiment

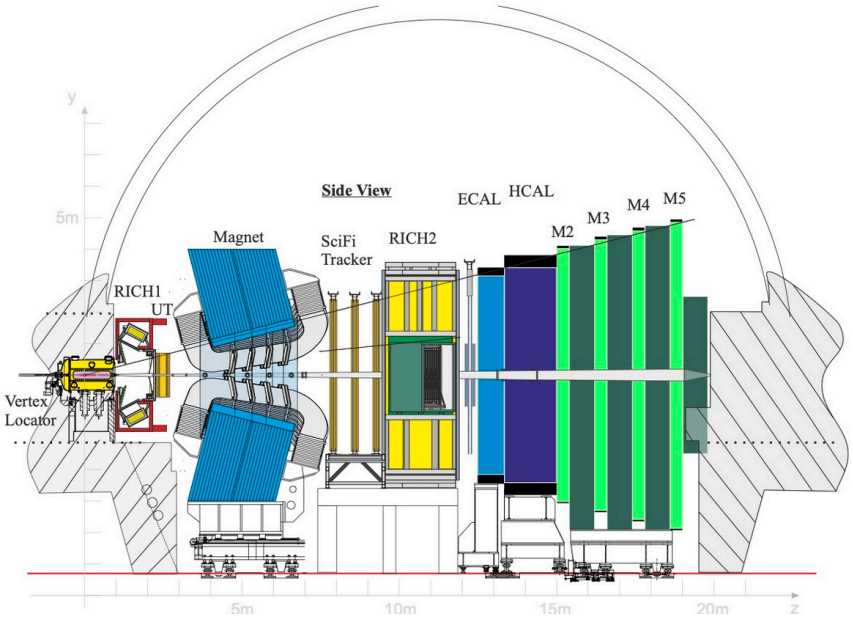
Tsinghua University

6th International Workshop on Future Tau Charm Facilities

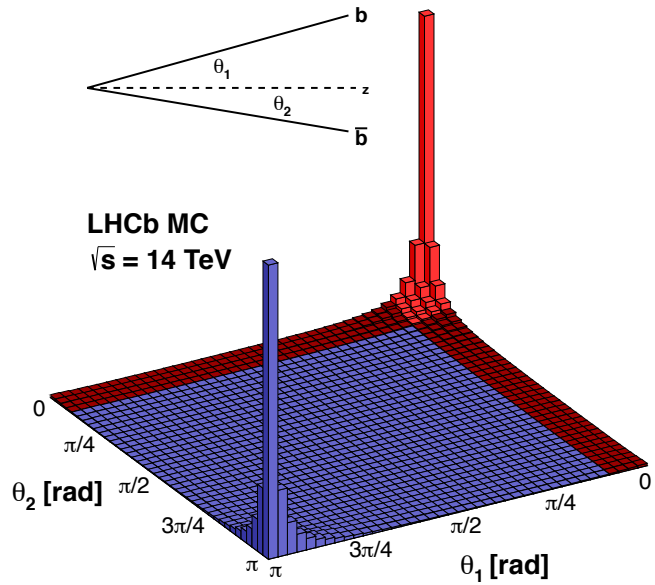
The LHCb Experiment

The LHCb Experiment

- LHCb is a single-arm spectrometer in the forward region ($2 < \eta < 5$) at the LHC: optimised for studying b (and c) decays boosted in the forward region. JINST 19 (2024) 05, P05065
- Flexible enough to support a broad range of physics: heavy ions and massive bosons (Z, W, H) results are competitive in the forward region and discoveries of charmed exotics in b decays.

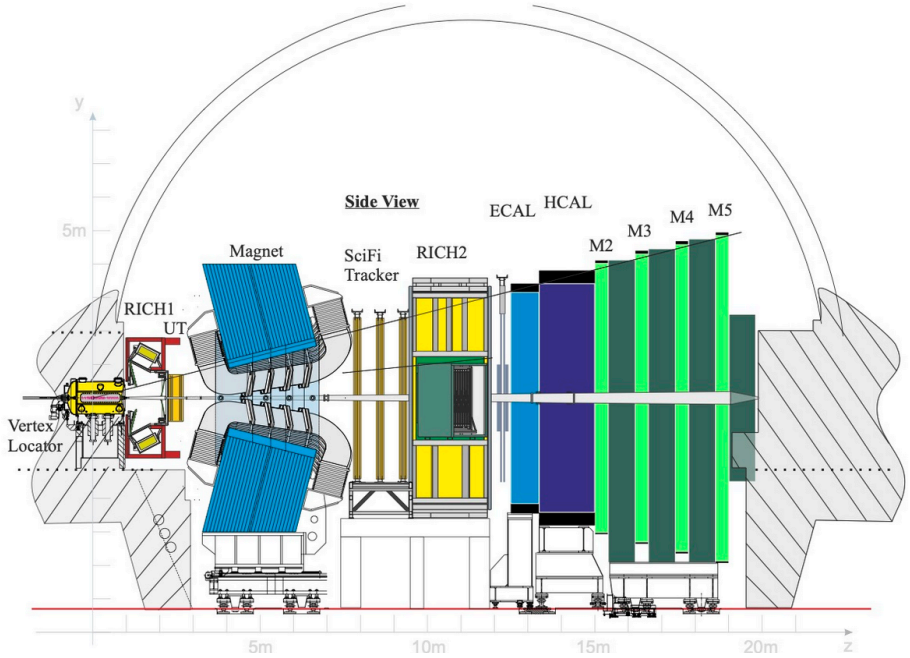


LHCb cross-section



Boosting of $b\bar{b}$ pair

The Current Run 3 LHCb Detector

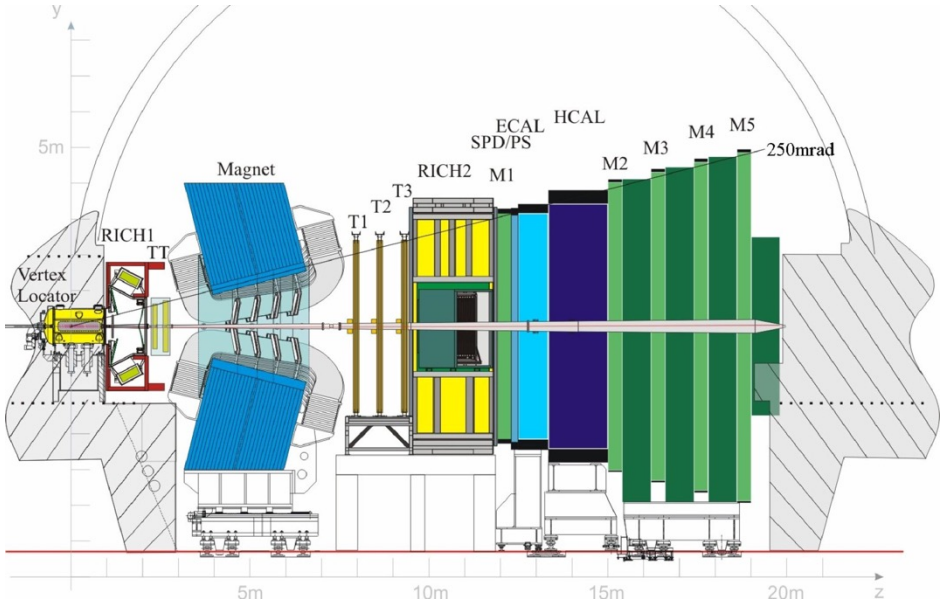


- VELO, UT and SciFi trackers.
  - LHCb originally designed for flavor physics and *CP* violation: require good momentum, mass and vertex resolution.
  - **VE**rtex **LO**cator - $55\mu m \times 55\mu m$ silicon pixel detector.
  - **U**pstream **T**racker - silicon strip detectors before magnet.
  - **Sci**tilating **Fi**bre - scintillators with SiPM readout after magnet.
- Muon, RICH, ECAL and HCAL detectors for particle identification.

Run 2 LHCb Detector

- Run 3 detector is designed achieve performance equal to or better than Run 2 detector (figure on right).

Int.J.Mod.Phys.A 30 (2015) 07, 1530022

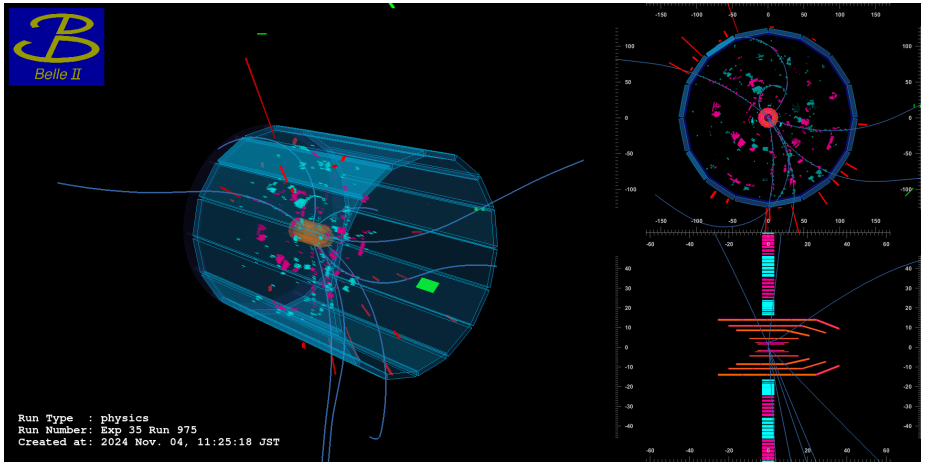


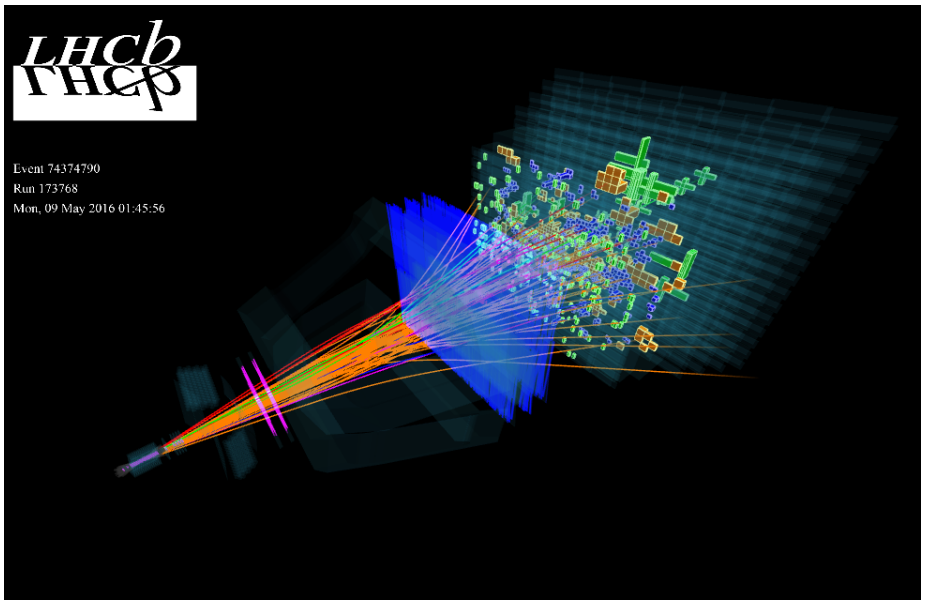
- Run 2 tracking performance:
  - Momentum resolution $\sigma_p/p \approx 0.5 - 1\%$.
  - IP resolution $(15 + 29/p_T\ [\text{GeV}])\mu m$.
- Run 2 PID performance:
  - $\epsilon(e) \sim 90\%$, $e \to h$ misID $\sim 5\%$.
  - $\epsilon(K) \sim 95\%$, $\pi \to K$ misID $\sim 5\%$.
  - $\epsilon(\mu) \sim 97\%$, $\pi \to \mu$ misID $\sim 1 - 3\%$.

Real Time Analysis at 30MHz

Data-taking in e^+e^- and Hadron Colliders

- e^+e^- is relatively clean.
- *Belle II* online event display on 4th November. [link]

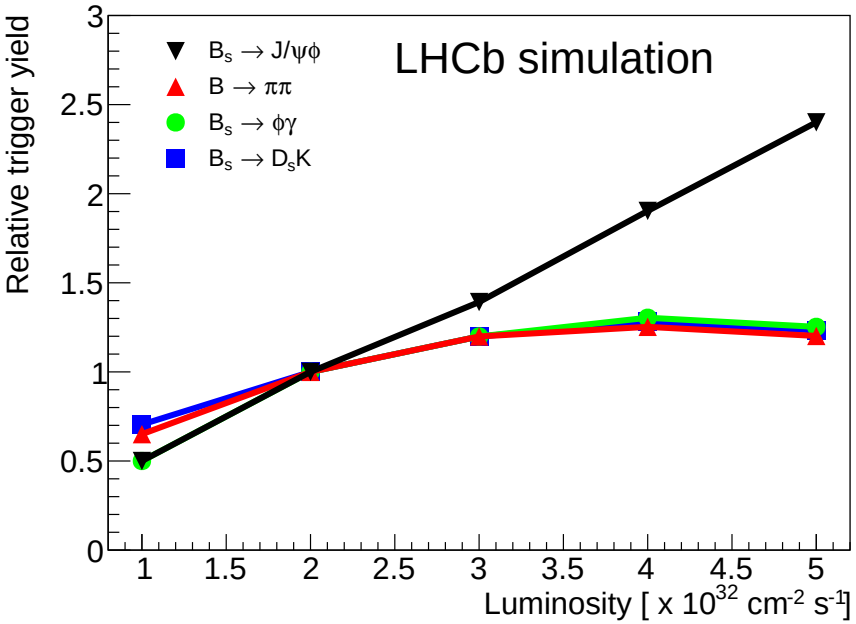




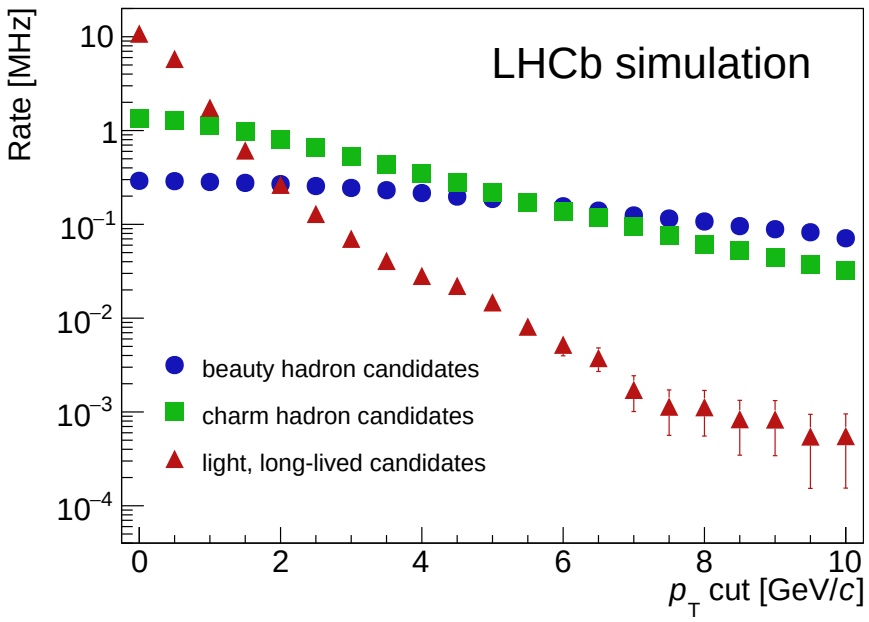
- In contrast, LHCb events are very busy.
- Typical LHCb event in Run 2. OPEN-PHO-EXP-2016-007

Physics Saturation with Hardware Triggers

- During LHCb Run 2, the L0 hardware trigger saturates at $\mathcal{L}_{\text{inst}} = 4 \times 10^{32}$ cm^{-2} s^{-1}.*
- At $5 \times \mathcal{L}_{\text{Run 2}}$, Run 3 cannot operate a hardware trigger without sacrifices to physics efficiencies. JINST 19 (2024) 05, P05065



b Triggers Saturation



Yields vs p_T

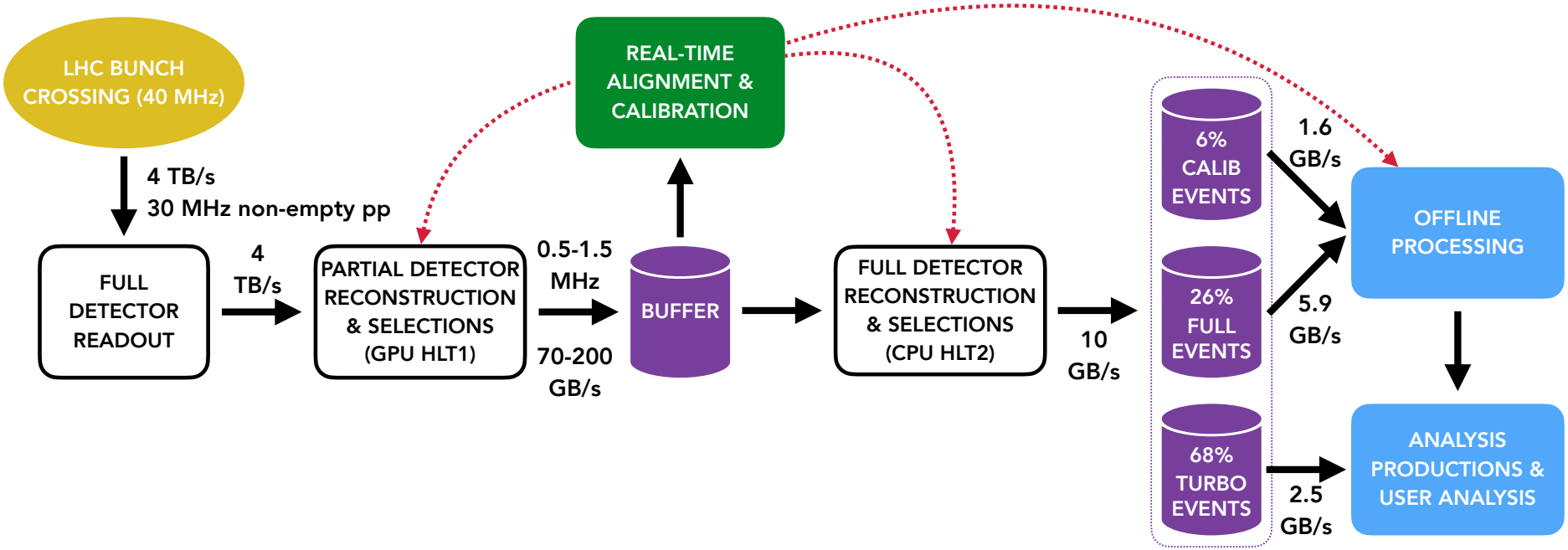
*Low occupancy in muon detectors → decays with μ have lower thresholds.

Real Time Analysis at 30MHz

- Problems: LHCb-PUB-2014-027
  - Trigger at $5 \times \mathcal{L}_{\text{Run 2}}$ without saturating physics efficiency.
  - Run 3: every 53.5 events have a *b*-hadron, 4.7 for *c*-hadron.

Real Time Analysis at 30MHz

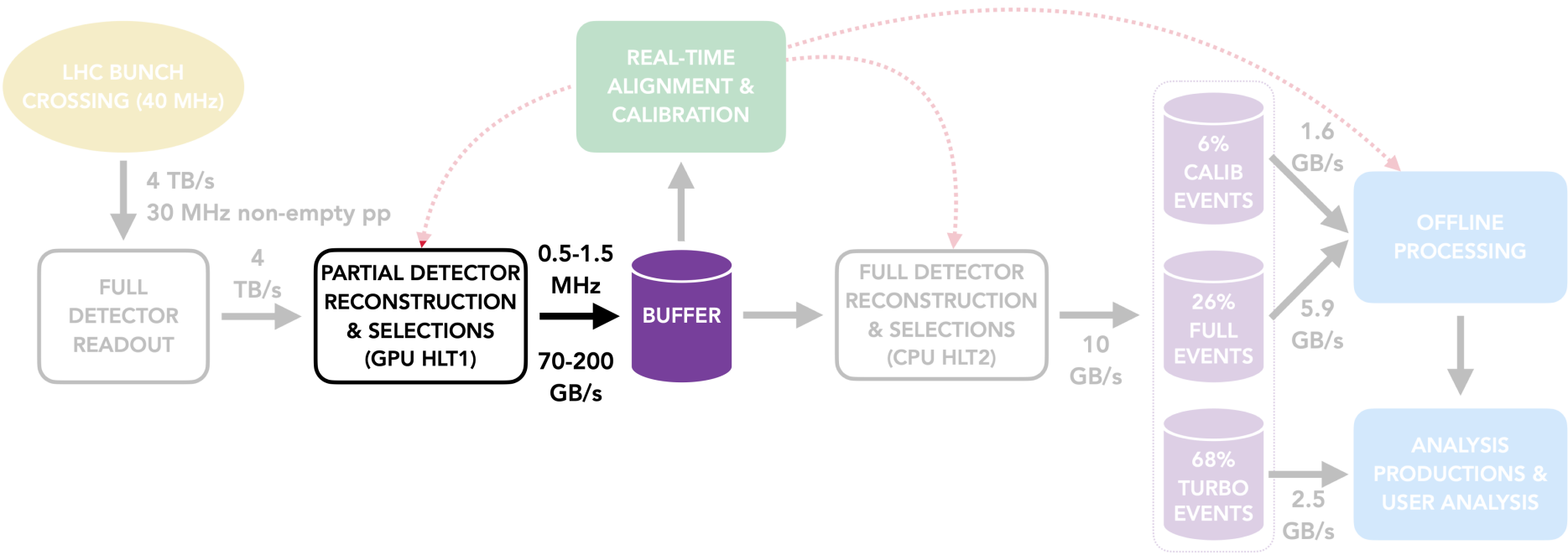
- Problems: LHCb-PUB-2014-027
  - Trigger at $5 \times \mathcal{L}_{\text{Run 2}}$ without saturating physics efficiency.
  - Run 3: every 53.5 events have a b-hadron, 4.7 for c-hadron.
- Solution: real-time software trigger that reconstructs full detector readout of 4TB/s and selects events at 30MHz.



JINST 19 (2024) 05, P05065

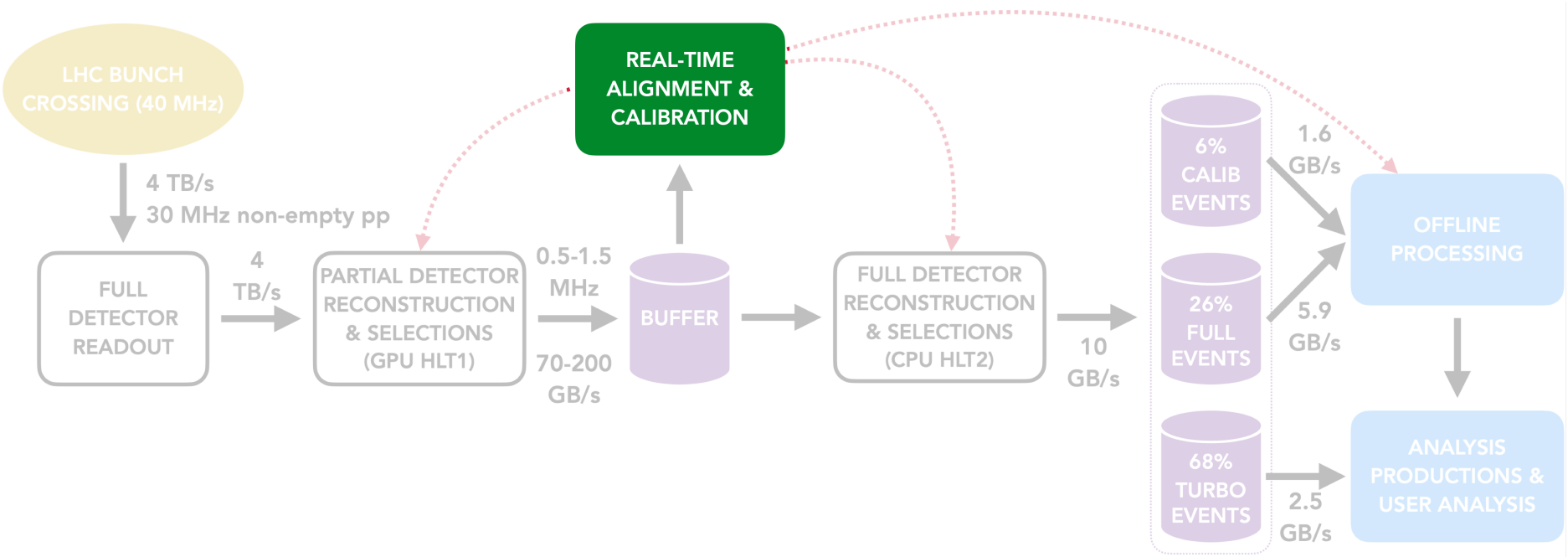
Building Blocks of Real Time Analysis at 30MHz

- Allen: high throughput inclusive `HLT1` trigger. Focus of this talk!



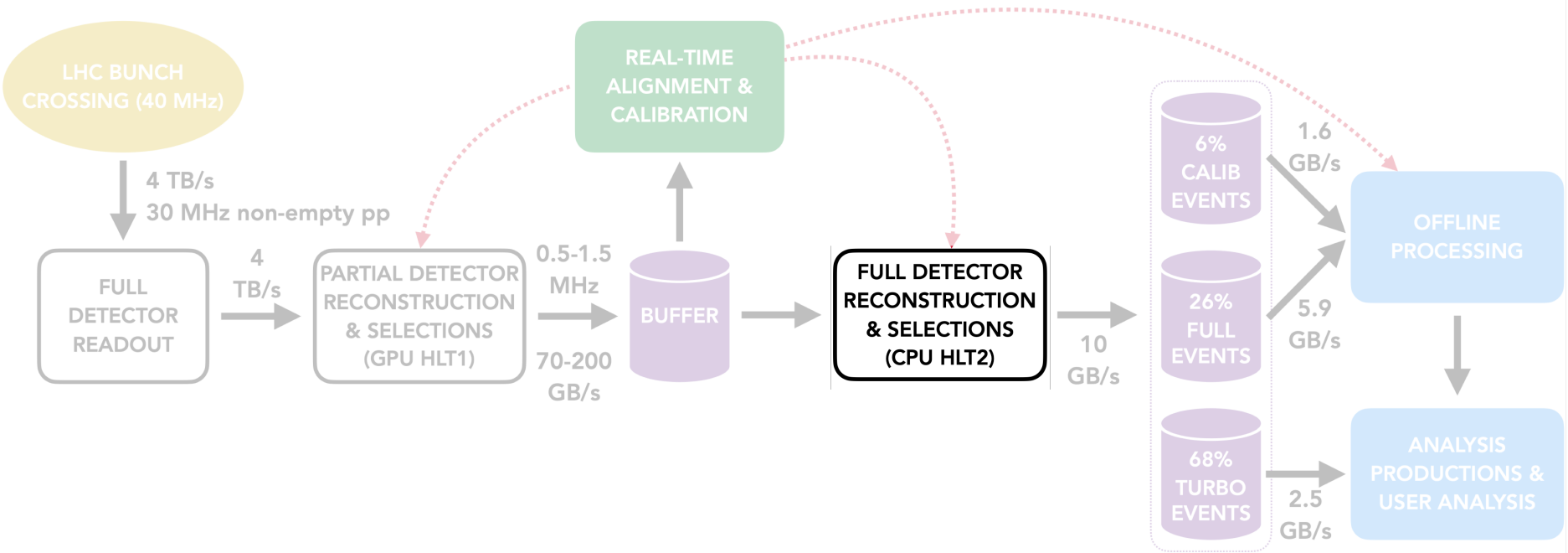
Building Blocks of Real Time Analysis at 30MHz

- Allen: high throughput inclusive `HLT1` trigger. Focus of this talk!
- Real-time alignment and calibration for offline quality detector alignment during online reconstruction.



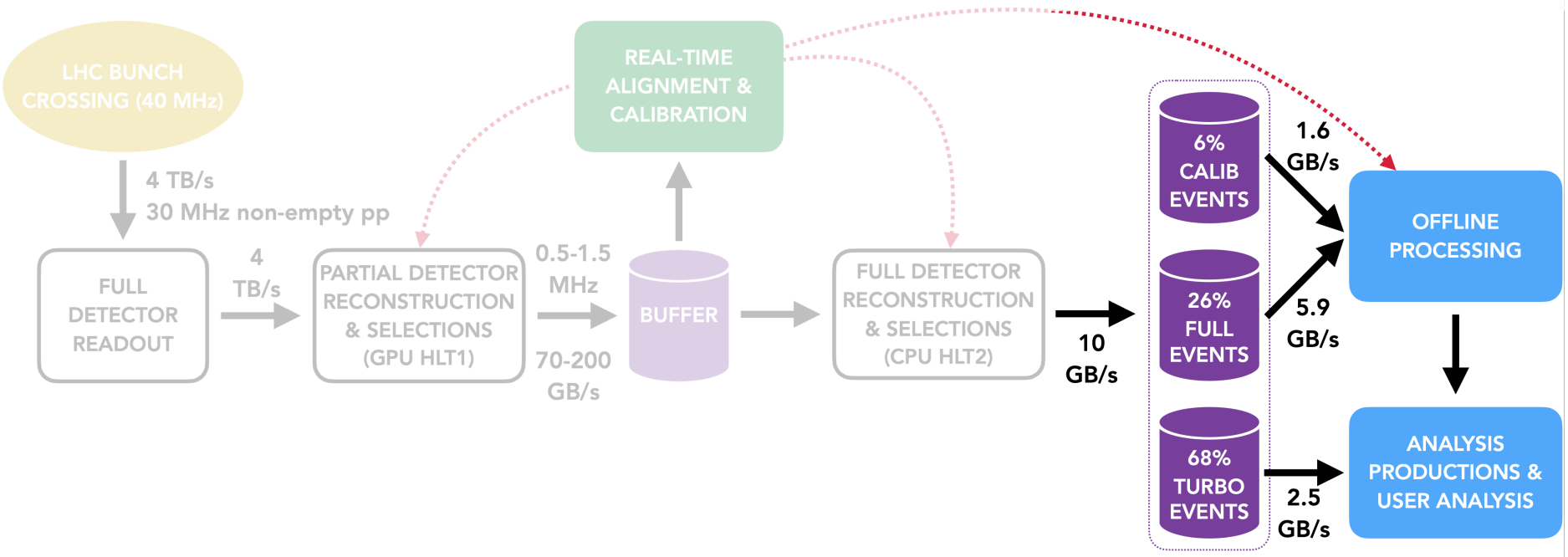
Building Blocks of Real Time Analysis at 30MHz

- Allen: high throughput inclusive `HLT1` trigger. Focus of this talk!
- Real-time alignment and calibration for offline quality detector alignment during online reconstruction.
- Offline quality reconstruction and flexible inclusive triggers at `HLT2` to support broad physics program.



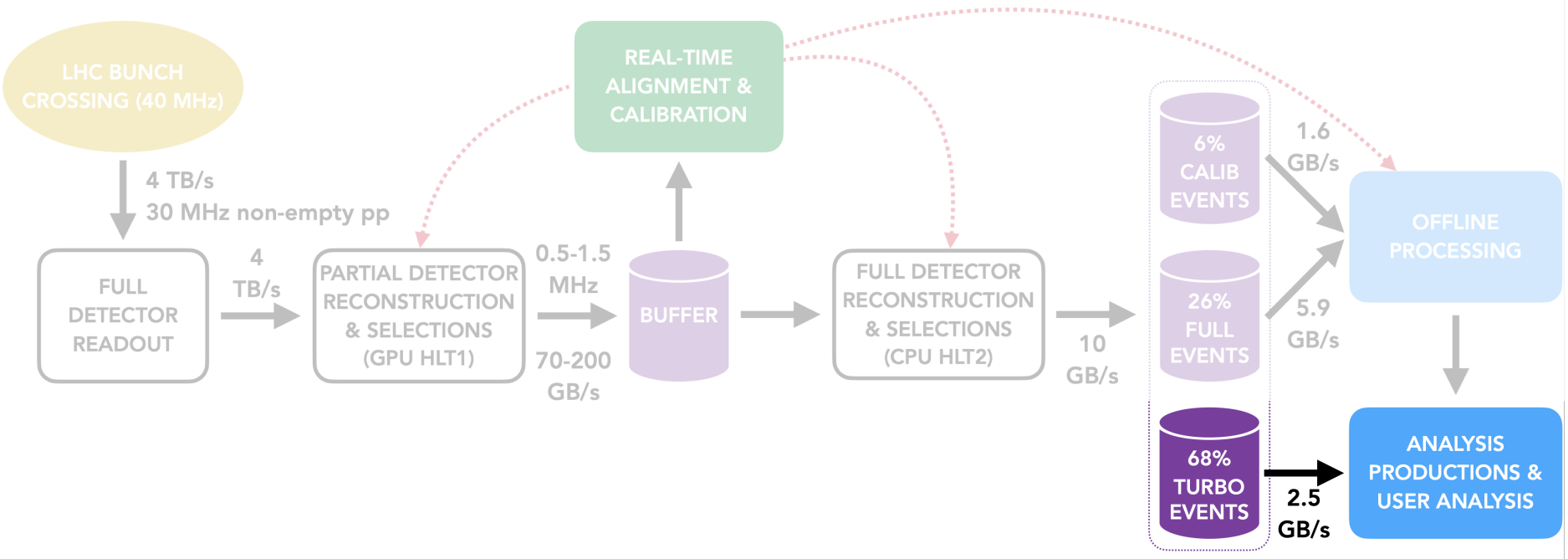
Building Blocks of Real Time Analysis at 30MHz

- Allen: high throughput inclusive `HLT1` trigger. Focus of this talk!
- Real-time alignment and calibration for offline quality detector alignment during online reconstruction.
- Offline quality reconstruction and flexible inclusive triggers at `HLT2` to support broad physics program.
- Multi stream storage.



Building Blocks of Real Time Analysis at 30MHz

- Allen: high throughput inclusive `HLT1` trigger. Focus of this talk!
- Real-time alignment and calibration for offline quality detector alignment during online reconstruction.
- Offline quality reconstruction and flexible inclusive triggers at `HLT2` to support broad physics program.
- Multi stream storage. TURBO persists partial event information to reduce data bandwidth. JINST 14 (2019) P04006



Allen: GPU Trigger at the LHCb Experiment

Allen Framework

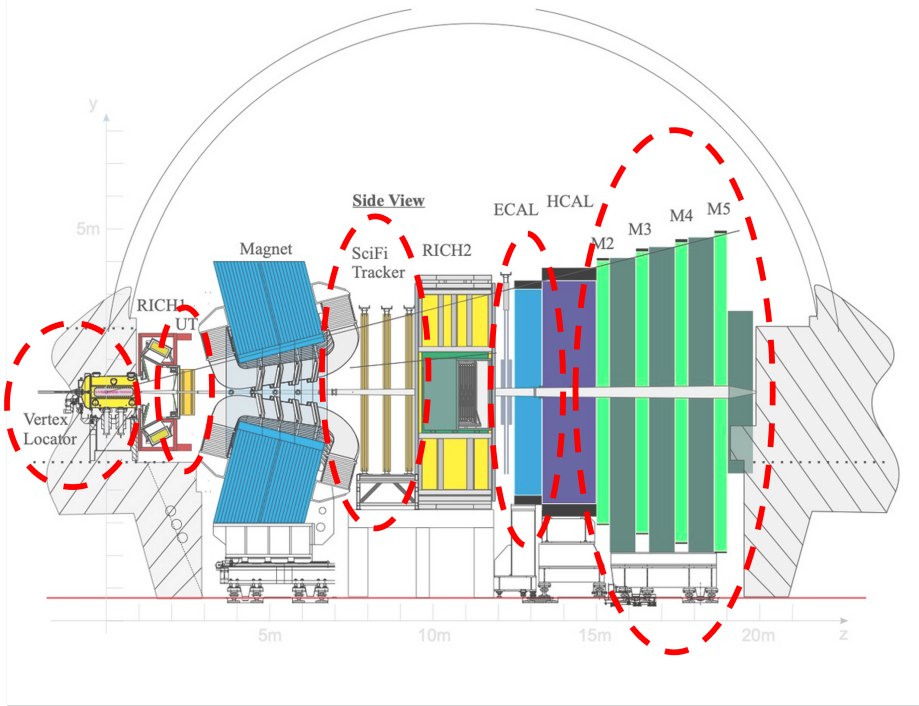
- GPU based trigger - Allen project, named after Turing Award computer scientist Frances Allen.
  - Computer Software for Big Science 4, 7 (2020)
  - gitlab and documentation

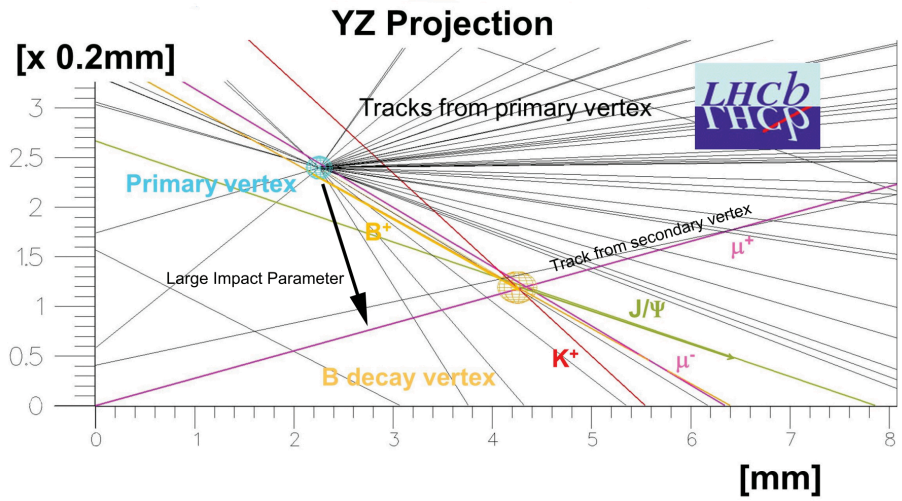
Allen Framework

- GPU based trigger - Allen project, named after Turing Award computer scientist Frances Allen.
  - Computer Software for Big Science 4, 7 (2020)
  - gitlab and documentation

- What are the physics that `Allen` needs to trigger on?
- How does `Allen` achieve a high throughput input event rate (30 MHz)?
- How does developers and analyst interact with `Allen`?

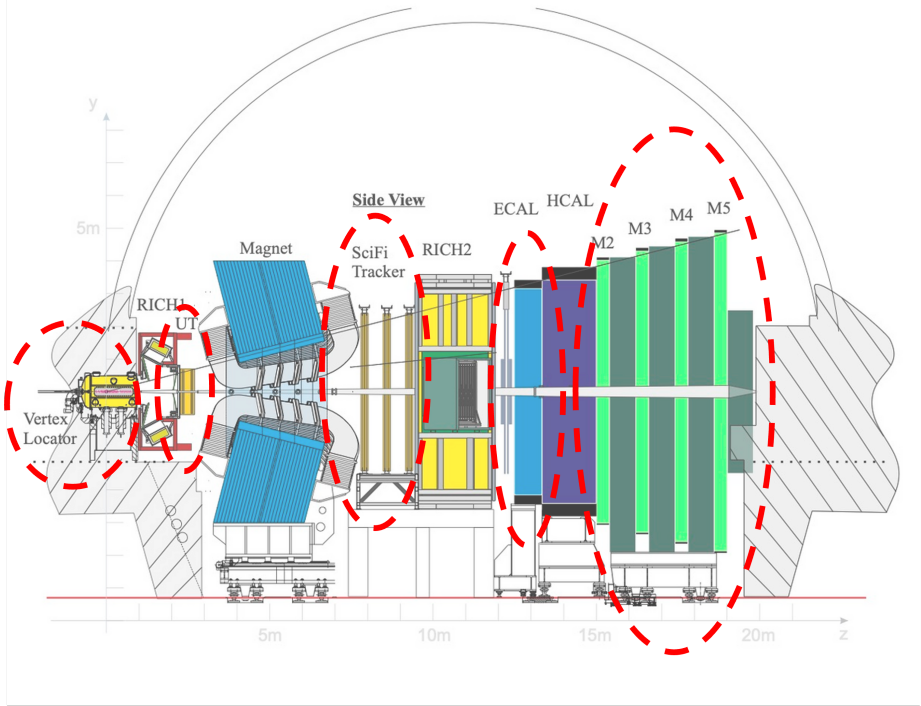
Allen Framework - Trigger Tasks

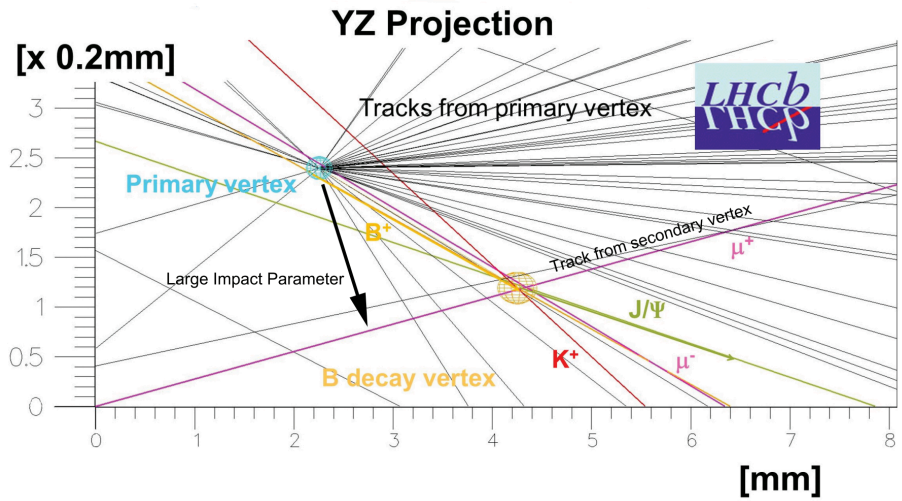




- Decode subdetectors that are reconstructed.
- Reconstruct event.
  - Reconstruct tracks.
  - Add particle identification information from `ECAL` and `MUON`.
  - Reconstruct vertices.

Allen Framework - Trigger Tasks





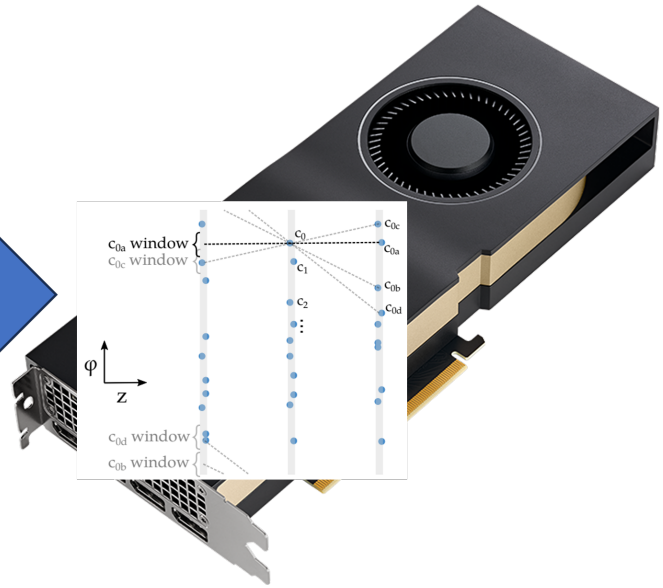
- Trigger inclusively on physics signature:
  1. Displaced secondary vertex and/or high momentum.
  2. Can relax cut with signature in ECAL or MUON detectors.
  3. ECAL reconstruction for γ.

Allen Framework - Processing Pipeline

Event Builders Servers

Raw Detector Data

Trigger Decisions

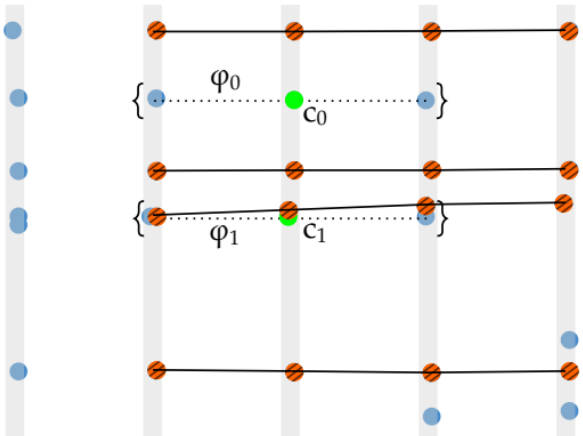


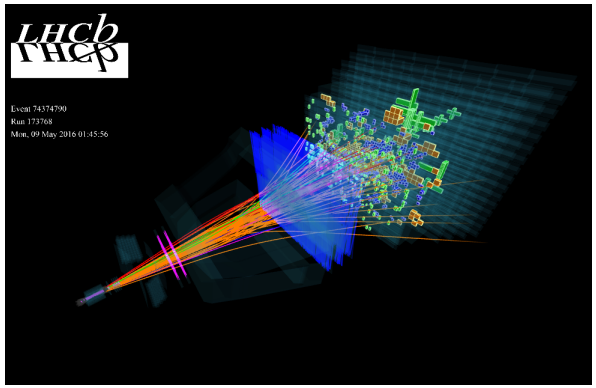
GPU

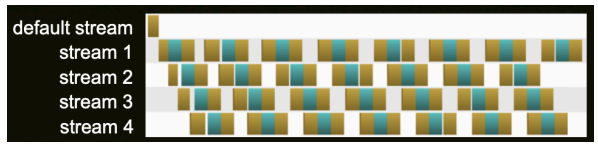
- Minimize data transfers between Event Builders and `Allen`.
  - Send raw detector data to GPUs over `PCIE40`.
  - `Allen` sends back trigger decisions - very little data.

Allen Framework - Hierachical Parallelism

- GPUs needs to process a lot of data in parallel to achieve high throughput.
- LHCb's HLT1 is inherently parallelizable:







Thread Level

- Parallelize within an event.
- $\mathcal{O}(100)$ tracks and raw banks, $\mathcal{O}(10)$ vertices, etc.

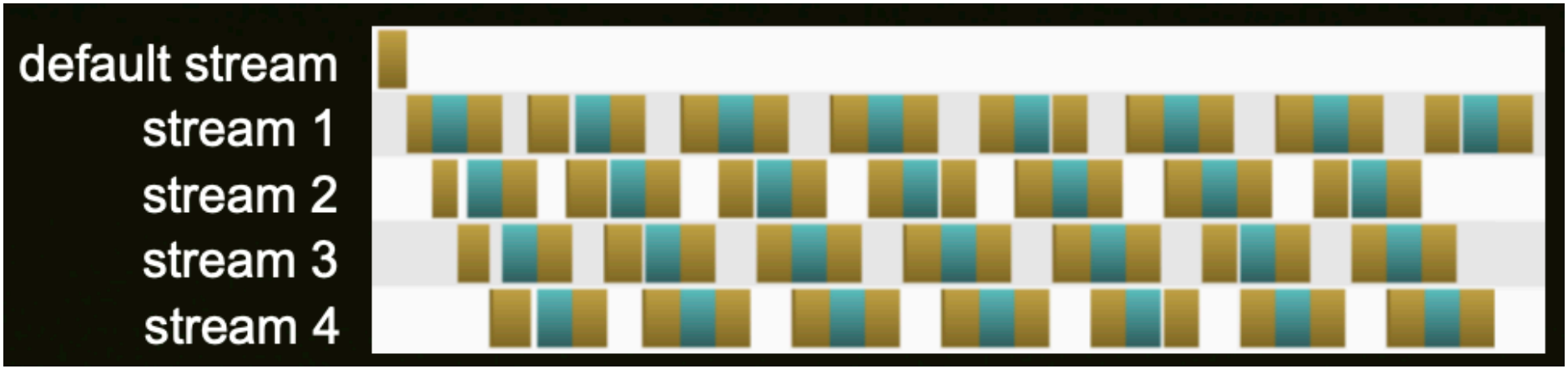
Block Level

- Parallelize over events.
- $\mathcal{O}(500)$ events per kernel call.

Stream/grid level

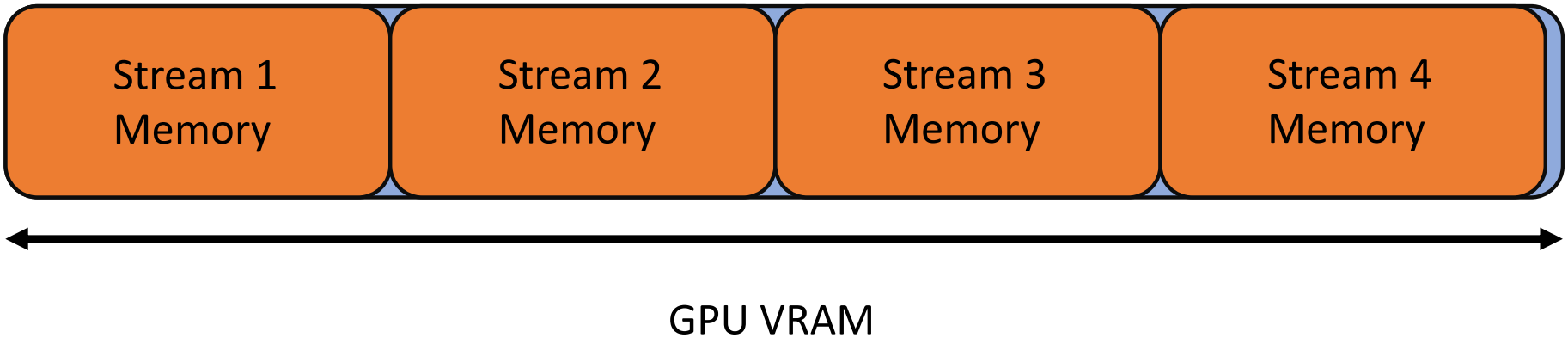
- $8 - 16$ CUDA streams per GPU.
- Parallelize over batches of events, typically $\mathcal{O}(500)$.

Allen Framework - CUDA Streams



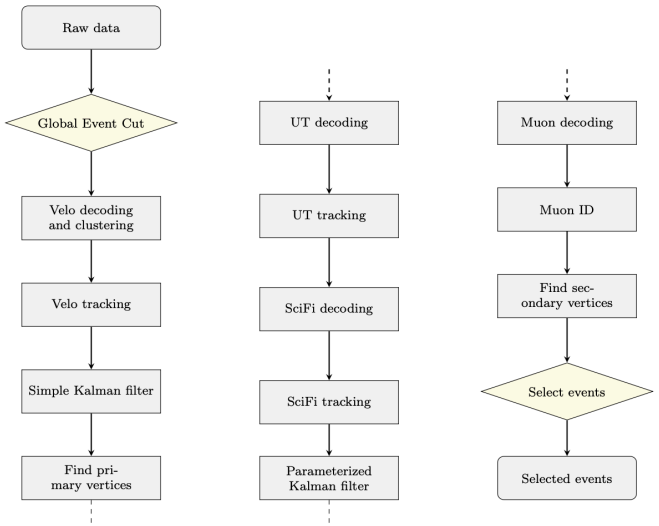
- Multiple batches of events are scheduled in parallel on GPUs.
  - Hide latency with asynchronous copy.
  - Hide latency of kernel call overhead.
  - Let CUDA scheduler on streaming multiprocessors optimize resource usage of GPUs.

Allen Framework - Memory Management



- Custom GPU memory manager.
  - Memory allocation and deallocation is slow → reduce throughput.
  - `Allen` allocates memory at start-up for each stream.
  - Algorithm request memory → manager dynamically returns chunk of unused pre-allocated memory.
  - Data dependency is used to track lifetime → dynamically returned unused chunk.

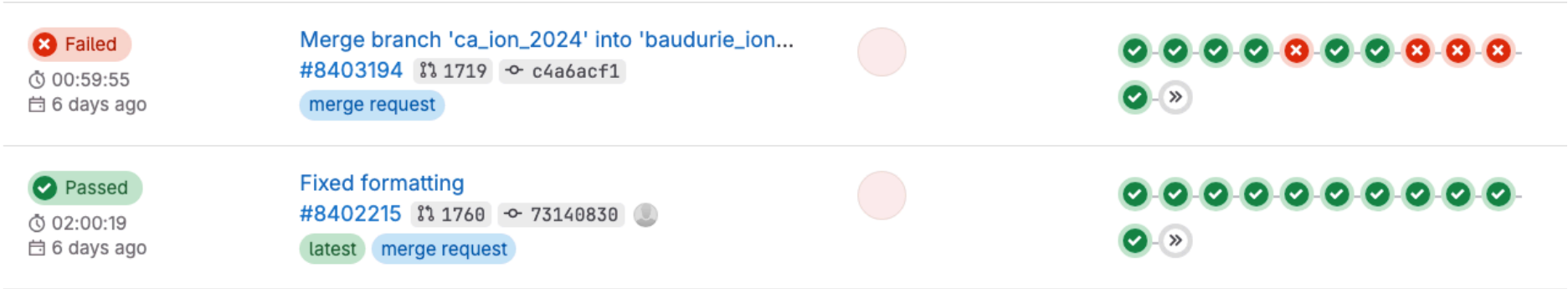
Allen Framework - Configuration



Example `Allen` sequence, each block is an algorithm.

- GPU algorithms are compiled into modular kernels.
- Configure sequence of kernels/algorithms in `python` using `LHCb`'s `PyConf` package.
- Flexibility to configure parameters, reconstruction logic and trigger lines for different data-taking conditions.
  - Different trigger configuration for *pp* vs heavy ion collision.

Allen Framework - Software



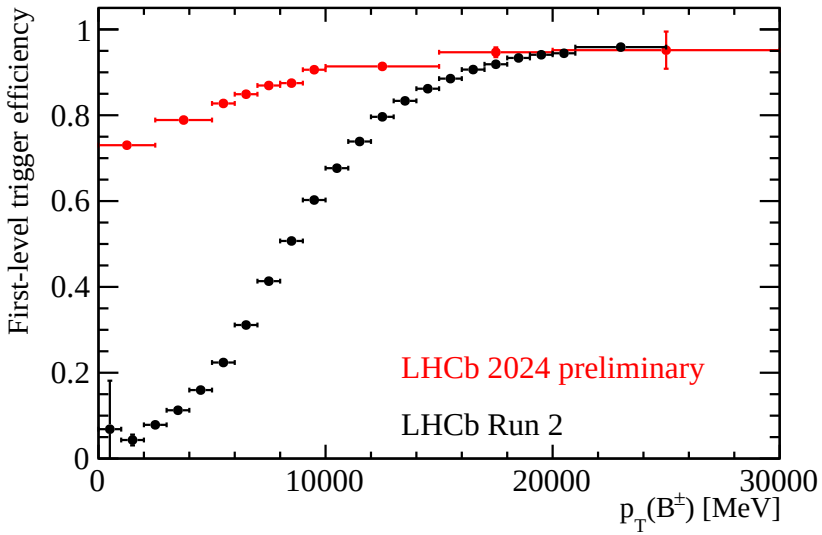
`Allen` CI tests.

- Good parallel programming requires different mindset.
  - `CUDA` very similar to `C++` but writing performant code is hard.
  - `Allen` has standalone build that is easy and fast to build.
  - Robust and fast `CI` pipeline for quick feedback to developers.
- `Allen` supports cross-architecture via macros.
  - Code written in `CUDA`, can compile to `NVIDIA GPU` and x86 `CPU`.
  - Architecture dependent code path for best performance.

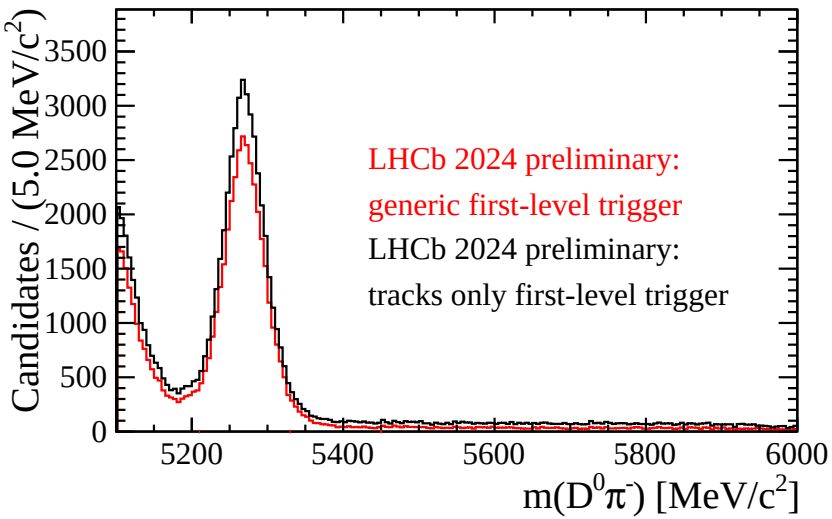
Data Taking with Allen

RTA Triggers - Real Data

- HLT1 trigger efficiency and mass resolution in $B^{\pm} \to D^0\pi^{\pm}$.
- Much better efficiency than Run 2.
  - Especially in low p_T where hardware triggers cut on.
- Good mass resolution with partial reconstruction of HLT1.



Efficiency

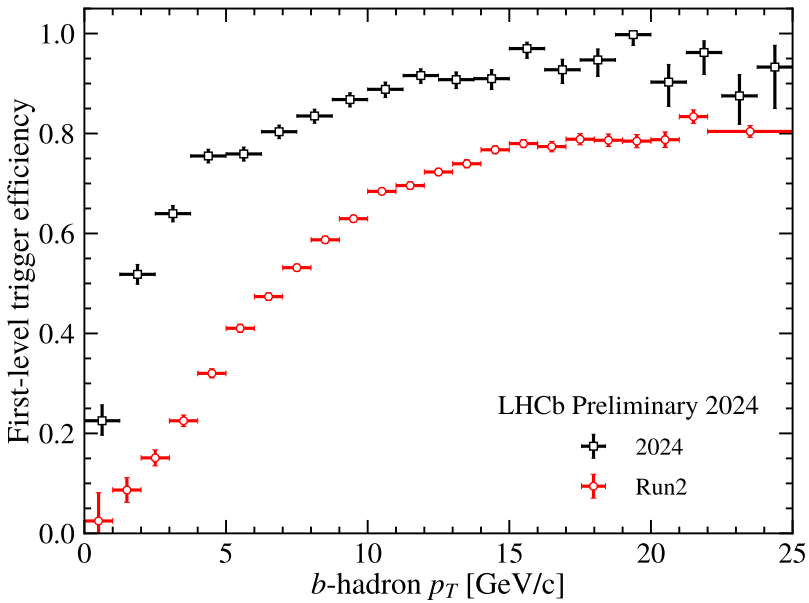


Resolution

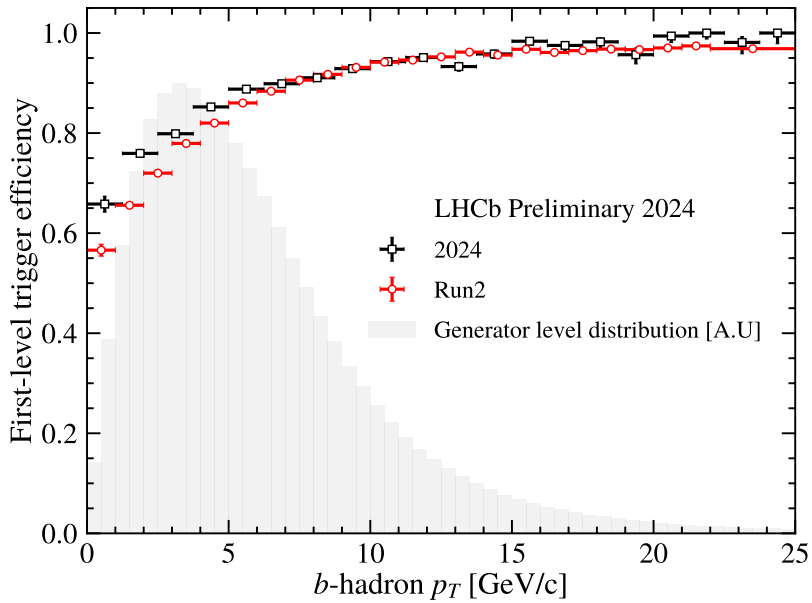
RTA Triggers - Real Data

- HLT1 trigger efficiencies in $B^{\pm} \to K^{\pm} J/\psi$, $J/\psi \to e^+e^-/\mu^+\mu^-$.
- Software trigger much more efficient for electrons.
- Small improvement with software triggers in muons.
  - The LHCb muon stations has low occupancy.
  - Loose but efficient muon hardware triggers in Run 2.

LHCB-FIGURE-2024-007



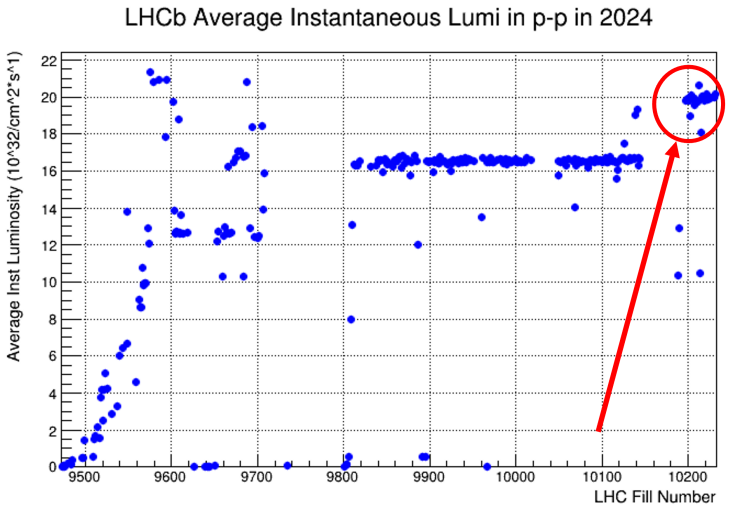
$J/\psi \to e^+e^-$



$J/\psi \to \mu^+\mu^-$

Allen - Design Luminosity

- Started taking data in 2022.
- First few months of data taking to fine tune `Allen` (GPU trigger) and real time analysis project.



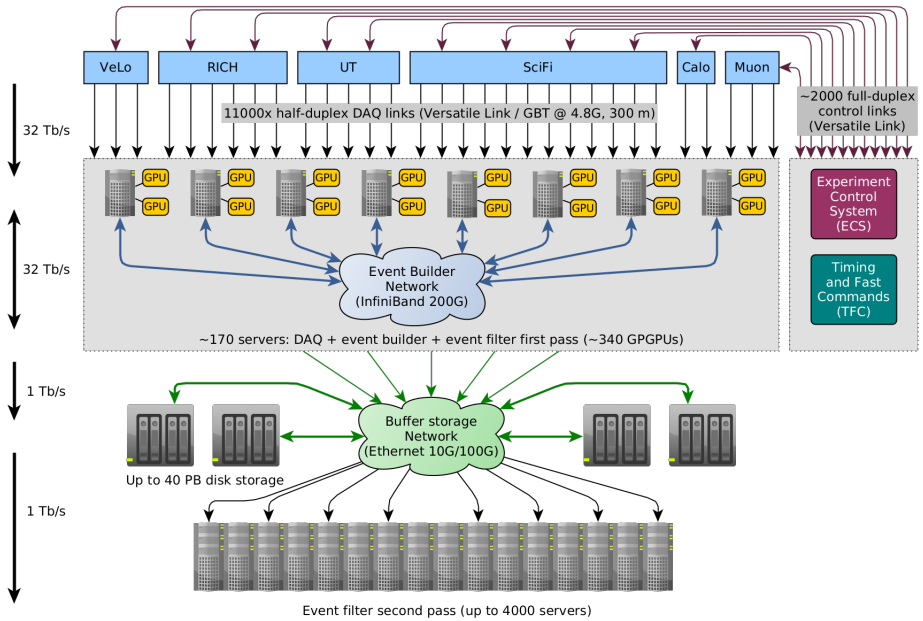
- 2024: successful data taking at design luminosity of $\mathcal{L}_{\text{inst}} = 2 \times 10^{33}\ \text{cm}^{-2}\ \text{s}^{-1}$.
- $\mathcal{O}(500)$ `NVIDIA A5000` GPUs processing at 4TB/s input rate.

Why Trigger with GPUs?

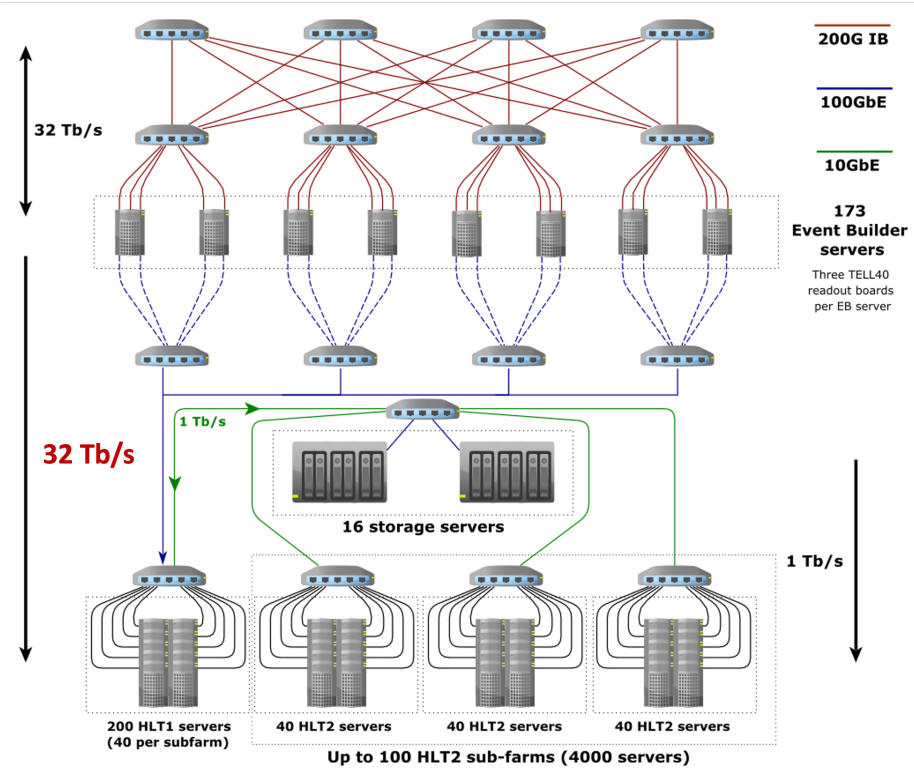
Data Acquisition Design

JINST 19 (2024) 05, P05065

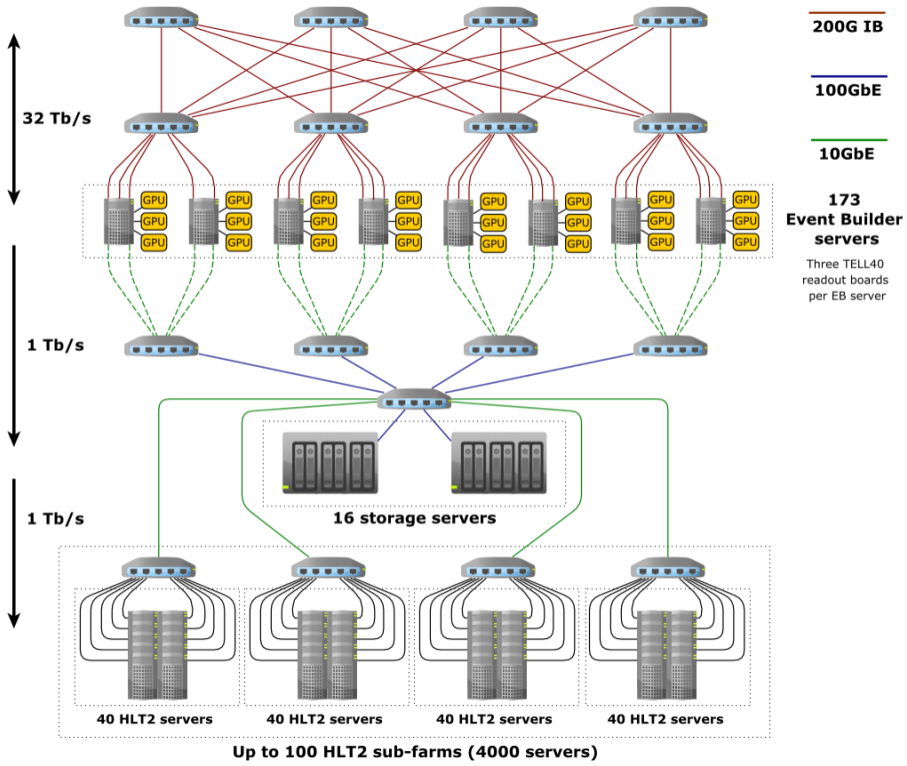
- $\mathcal{O}(500)$ FPGA receives detector data at 30 MHz, 4 TB/s.
- ~ 170 event builder (EB) node with 3 GPUs each to trigger with HLT1, output up to 200 GB/s.
- Store events in a buffer before HLT2 filter down to 10 GB/s.



What If HLT1 Is CPU Based?



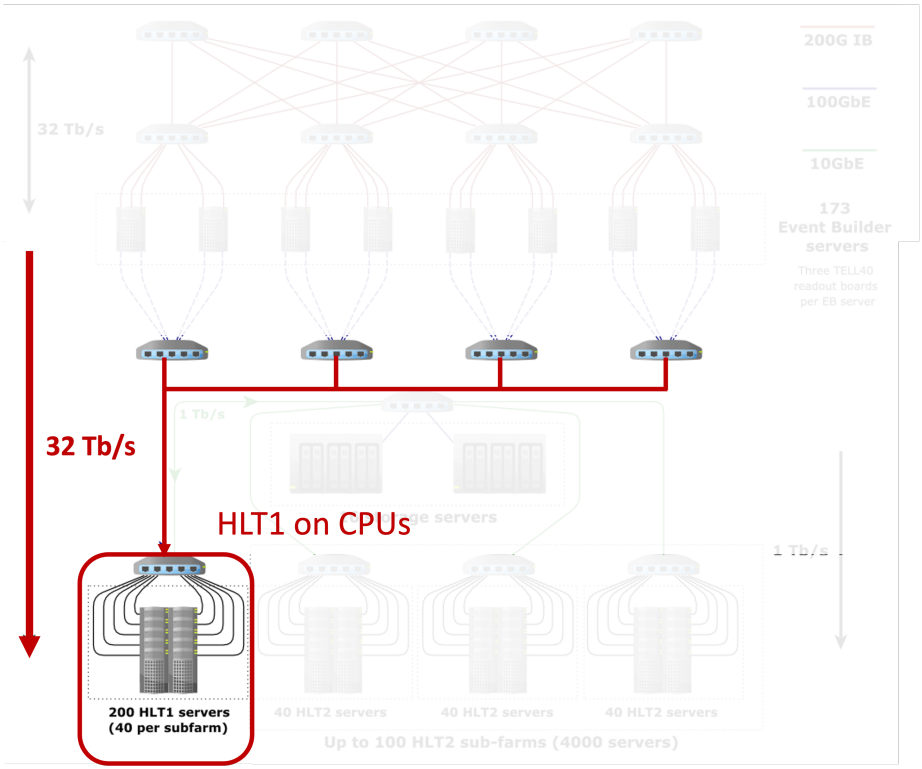
CPU based HLT1 DAQ



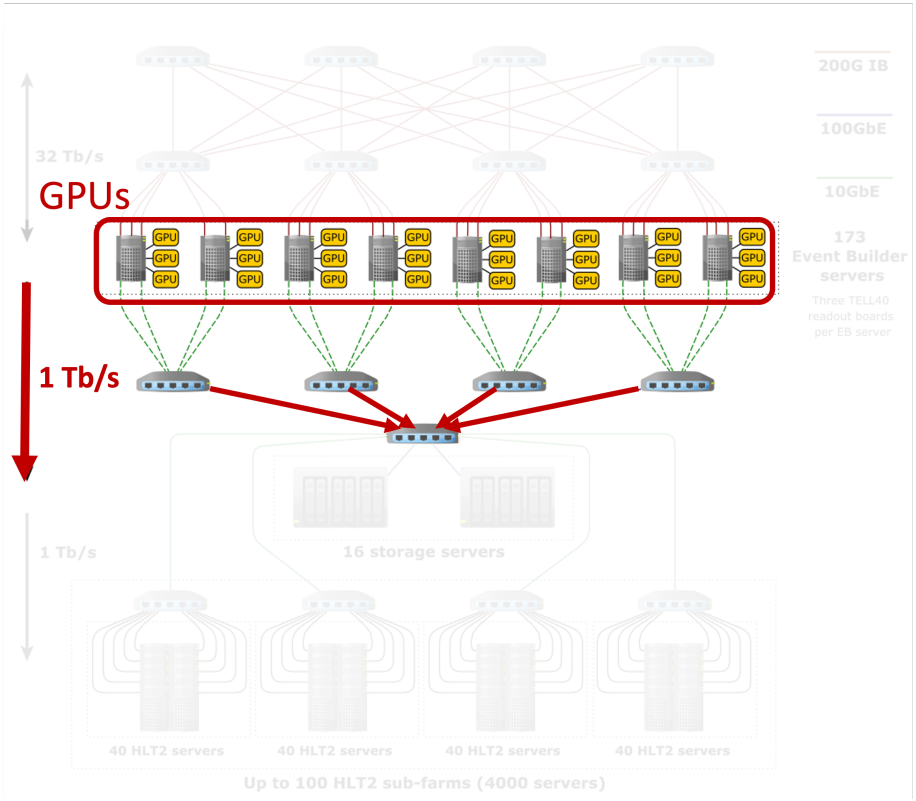
GPU based HLT1 DAQ

- This is too complicated, let's focus on the important part.

What If HLT1 Is CPU Based?



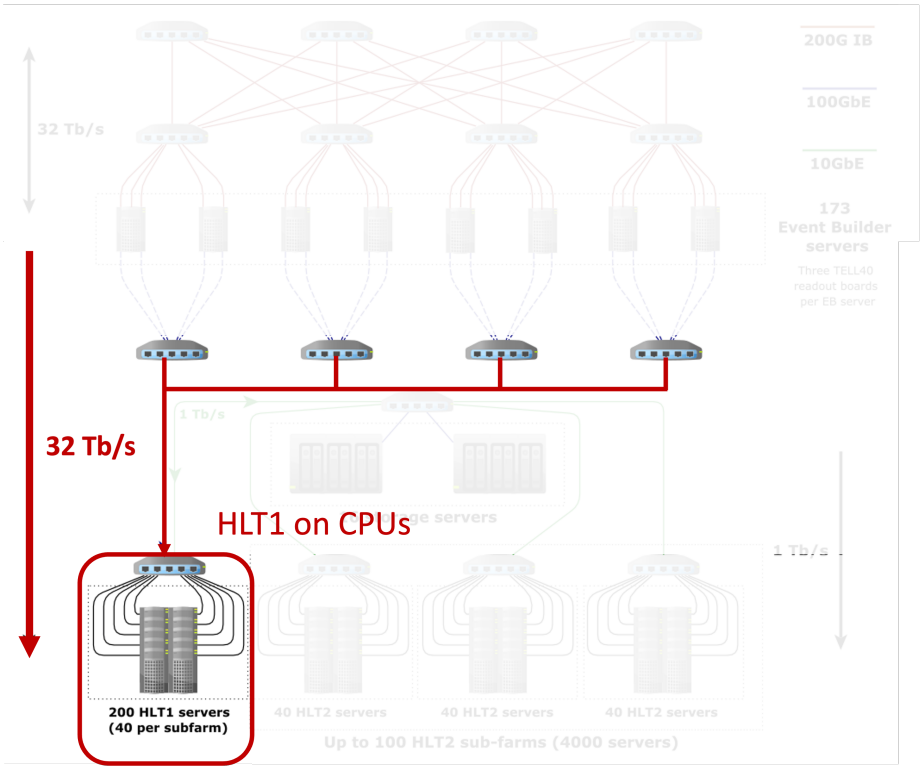
CPU based HLT1 DAQ



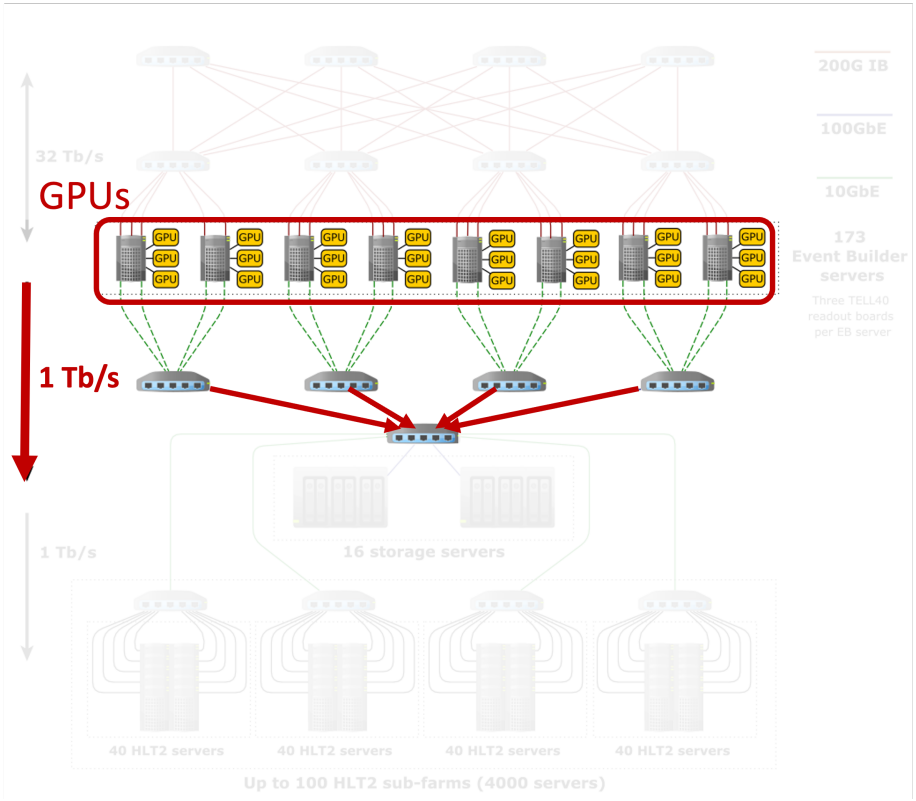
GPU based HLT1 DAQ

- GPUs are cheaper for the required throughput - 325k CHF.
- Also save costs on network - 250k CHF.

What If HLT1 Is CPU Based?



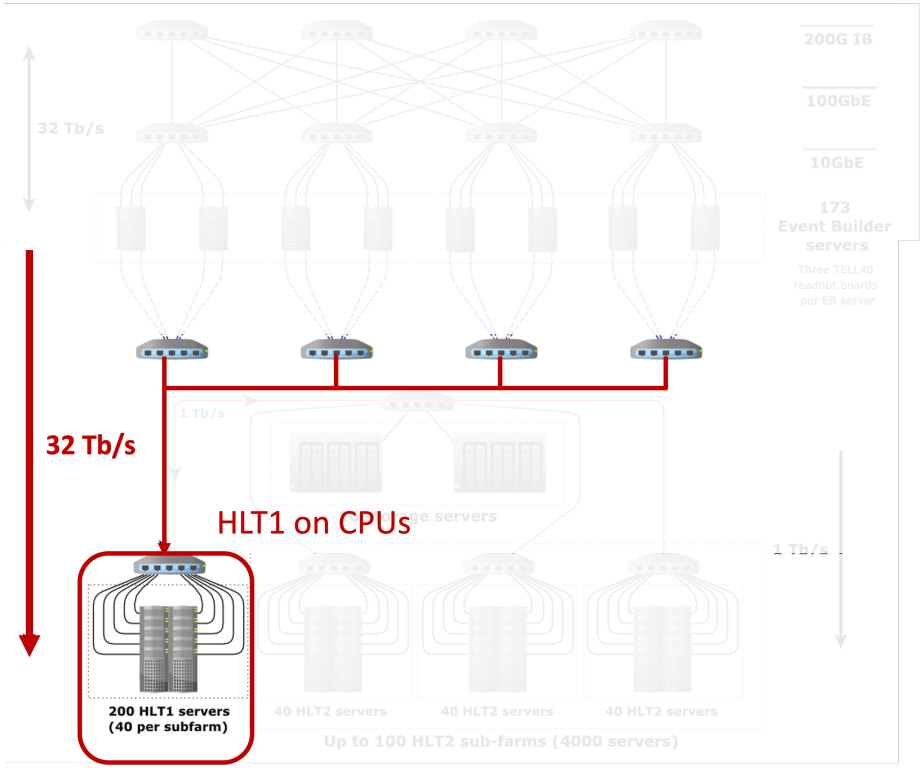
CPU based HLT1 DAQ



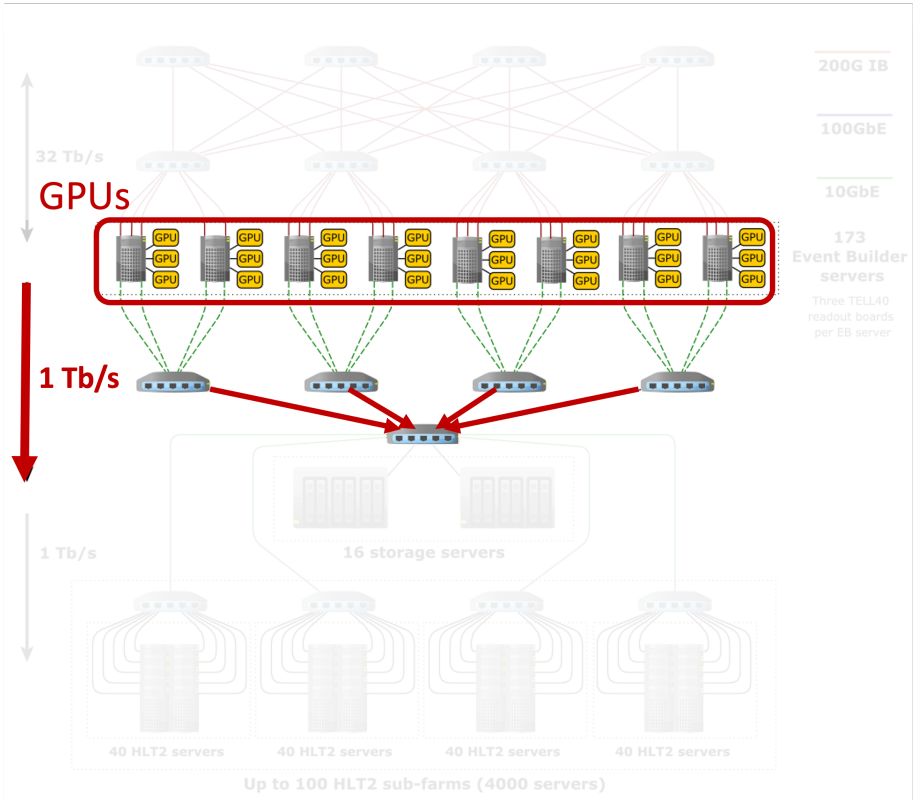
GPU based HLT1 DAQ

- GPUs are cheaper for the required throughput - 325k CHF.
- Also save costs on network - 250k CHF.
  - GPU: directly use PCIE40 on Event Builder (EB) nodes for free.
  - CPU: additional 32Tb/s EB-HLT1 connection. (4TB/s readout)

What If HLT1 Is CPU Based?



CPU based HLT1 DAQ



GPU based HLT1 DAQ

- Same physics efficiency but save 575k CHF, $\sim 20\%$ cost of filter farm.
  - Computing and Software for Big Science 6 (2022) 1, 1
  - Estimated in 2021, numbers might have changed by $\sim 10\%$.

Summary

Relevance for STCF

- GPU-based HLT1 is cheaper to CPU-based.
  - Needs enough parallelism for high throughput - I think this is doable in e^+e^- colliders but needs different implementation from LHCb.
  - Performant code needs different programming skills.
- HLT2 features:
  - Offline quality reconstruction: offline reconstruction uses same software.
  - Selective persistency: only save information relevant for analysis.
- Real time alignment and calibration
  - HLT1 and HLT2 reconstruction uses offline quality detector alignment and calibration.

Summary

- The Run 3 LHCb experiment, initially designed for b, c physics operates a fully software trigger.
  - Real-time analysis in the software trigger significantly improves efficiencies, even at higher luminosity of Run 3 data-taking.
  - Flexible design allows LHCb to support a broad range of physics program, heavy ion to heavy bosons.
- `Allen`, a GPU based trigger has been commissioned at the `LHCb` experiment.
  - High throughput data processing with lower costs than CPU solution.
  - The `Allen` framework is designed to run on GPUs.
  - Successfully taking data at design luminosity.

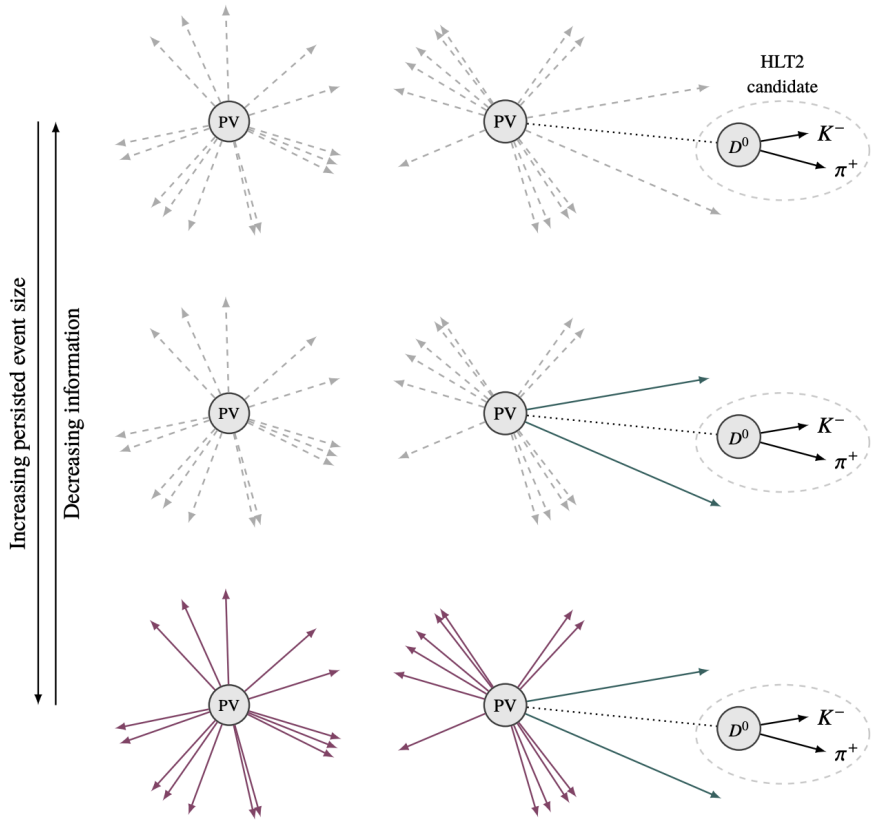
Backup

High Level Trigger 2 - Goals and Design

- Goal: Perform best possible reconstruction and reduce data rate to 10 GB/s.
- Complete pattern recognition, track fit, vertexing and PID.
- Currently have $\mathcal{O}(4000)$ trigger lines written by analysts for broad range of physics programs.
  - Structure of arrays to fully benefit from SIMD CPUs.
  - Throughput oriented (ThOr) functors agnostic to input/output type.
  - Functor cache instead of just-in-time compilation.
  - Event scheduler handles data dependencies with composite node (OR, AND, NOT). CERN-THESIS-2020-331

High Level Trigger 2 - Turbo Persistency

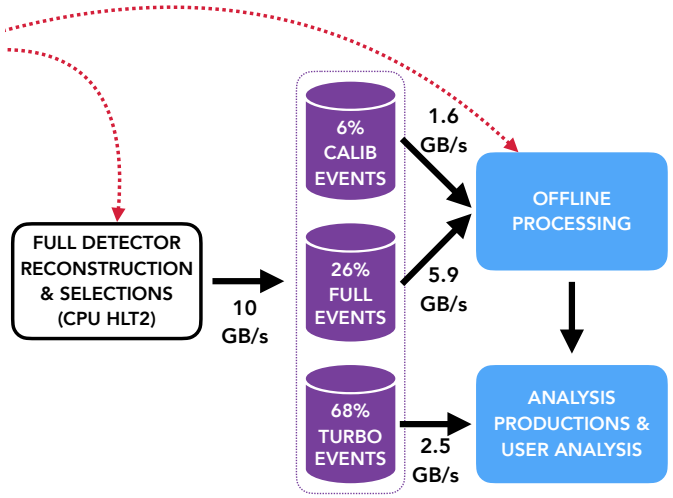
- HLT2 has 10 GB/s output but $\mathcal{O}(4000)$ trigger lines.
- Reduce event sizes by only saving information relevant to physics analysis.
- Flexible settings for analyst to choose what to persist.
- Used extensively in Run 2.



JINST 14 (2019) P04006

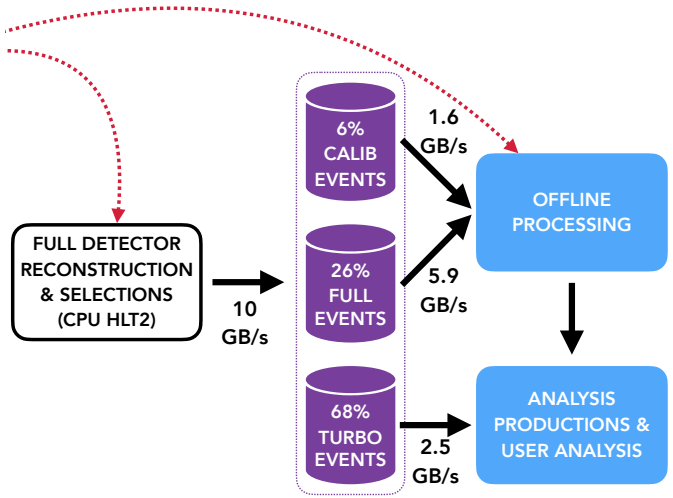
High Level Trigger 2 - Turbo Persistency

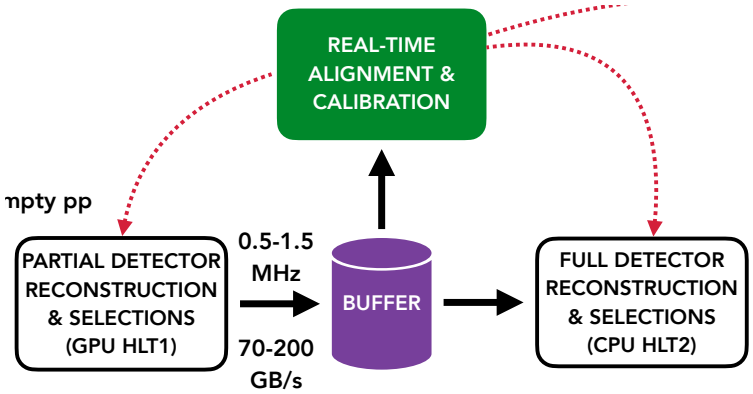
- Expand Turbo in Run 3.
  - Triggers with Turbo persistency account for $\sim 70\%$ of events triggered by LHCb.
  - But they only use 25% of HLT2 output data bandwidth!



High Level Trigger 2 - Turbo Persistency

- Expand Turbo in Run 3.
  - Triggers with Turbo persistency account for $\sim 70\%$ of events triggered by LHCb.
  - But they only use 25% of HLT2 output data bandwidth!

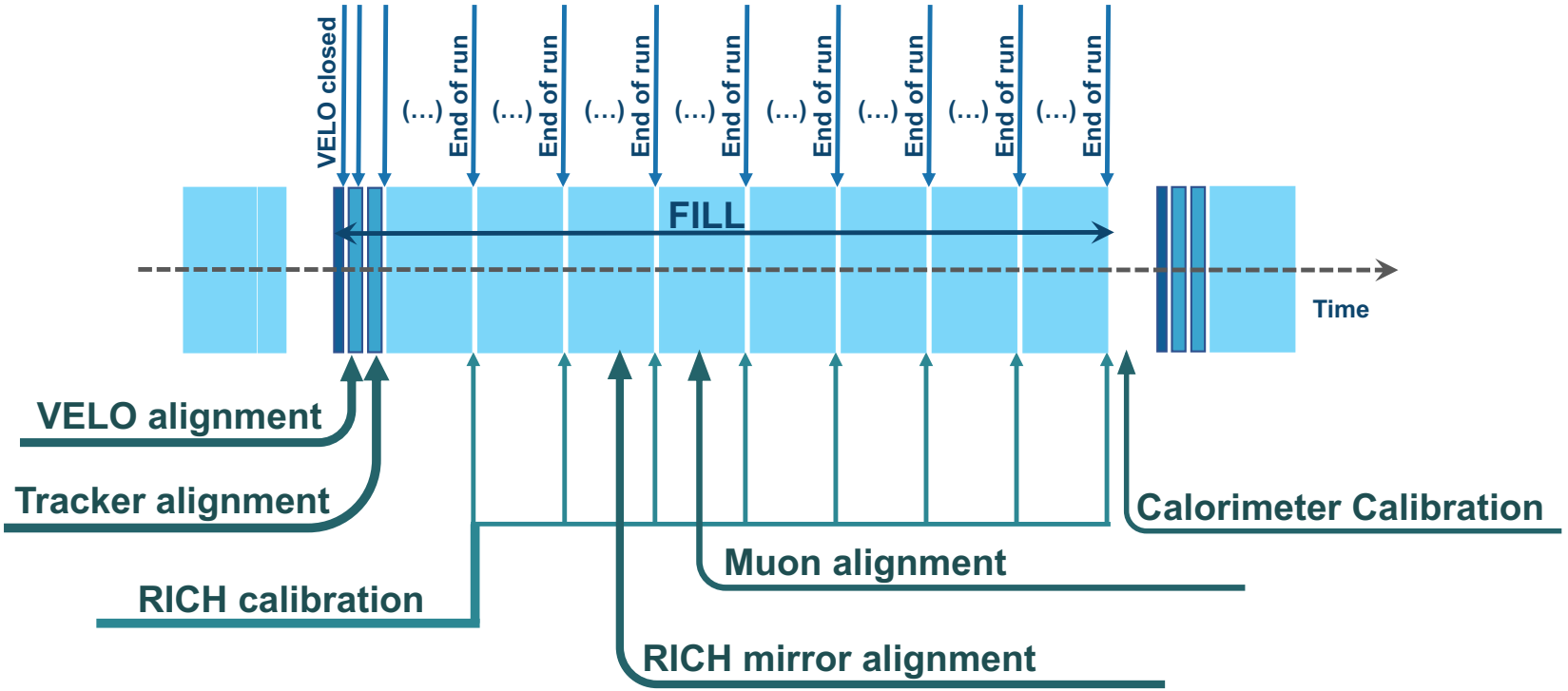




- Unlike full events, Turbo events cannot re-run reconstruction.
  - Need best possible alignment and calibration during HLT2.
  - Run alignment and calibration in real-time before HLT2.
  - HLT1 output is stored in a $\mathcal{O}(10)$ PB buffer.

Alignment and Calibration

- Real time alignment and calibration pioneered in Run 2.
  - Alignment for VELO, UT, SciFi, RICH mirrors and MUON.
  - Calibration for RICH, ECAL and HCAL.
- Critical for HLT2, especially Turbo selective persistency.



JINST 19 (2024) 05, P05065

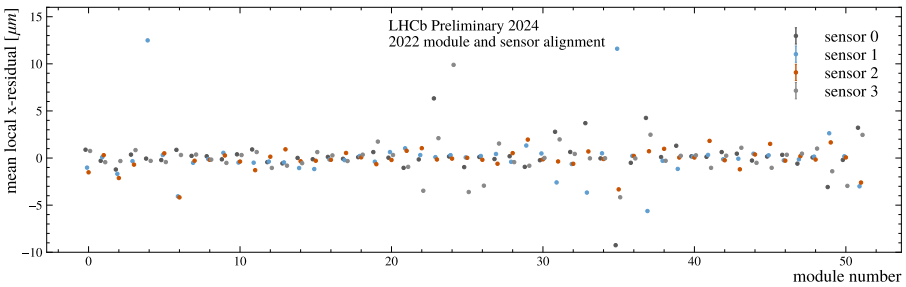
Alignment and Calibration - VELO

- Based on Analyzer and Iterator:
  - Analyzer runs reconstruction and calculates alignment constants.
  - Iterator updates it and checks for convergence.

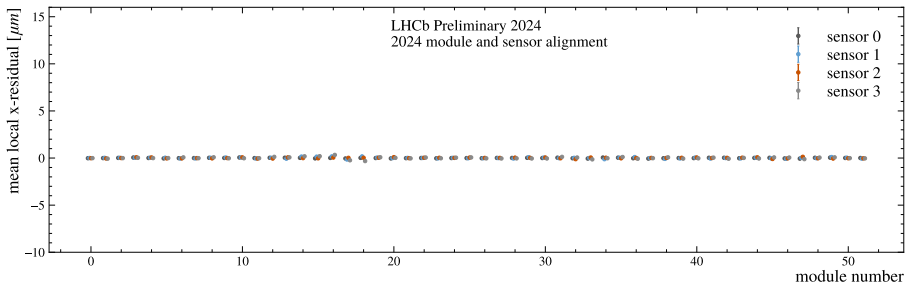
Alignment and Calibration - VELO

- Based on Analyzer and Iterator:
  - Analyzer runs reconstruction and calculates alignment constants.
  - Iterator updates it and checks for convergence.
- VELO alignment significantly reduces VELO hit residual.
  - Results in improved vertex resolution for triggers and physics analysis.

LHCB-FIGURE-2024-009



before

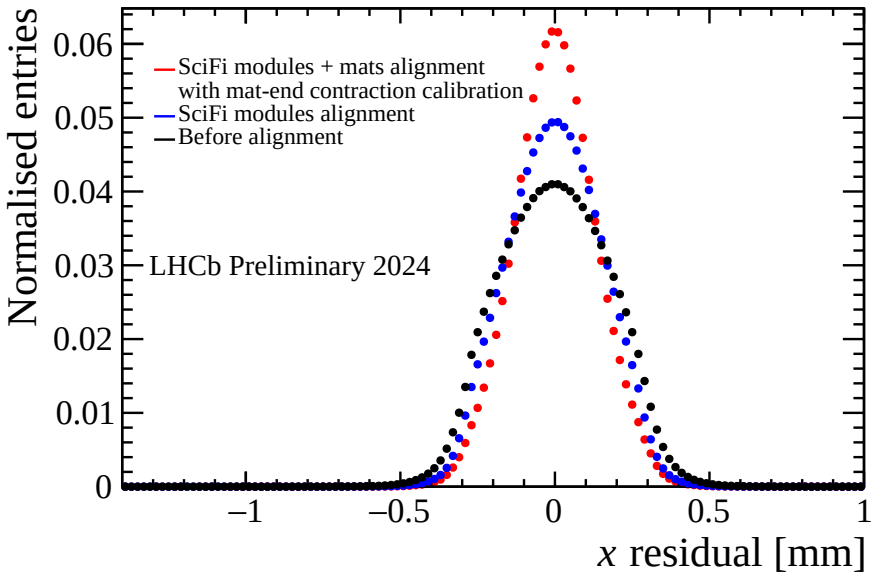


after

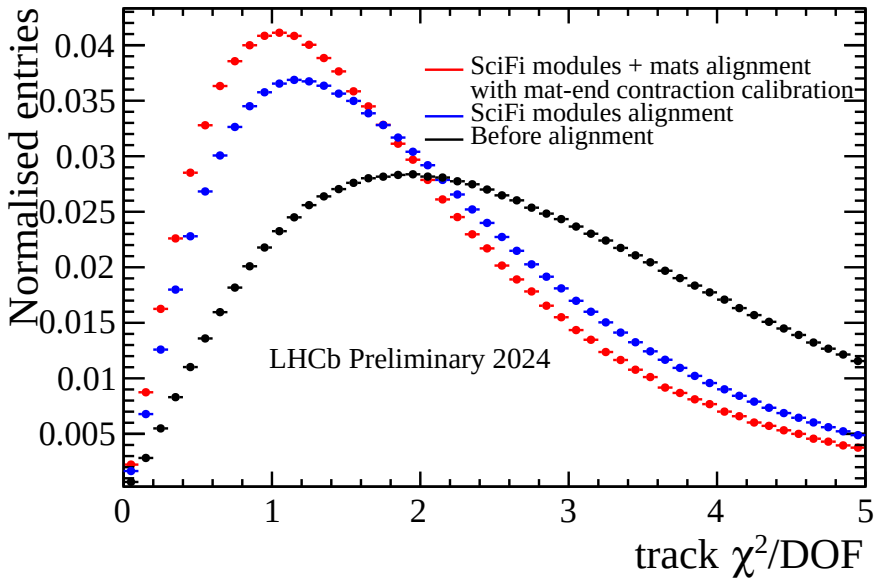
Alignment and Calibration - SciFi

- SciFi alignment significantly improves hit residual and track χ^2.
  - Results in improved momentum resolution.

LHCB-FIGURE-2024-009



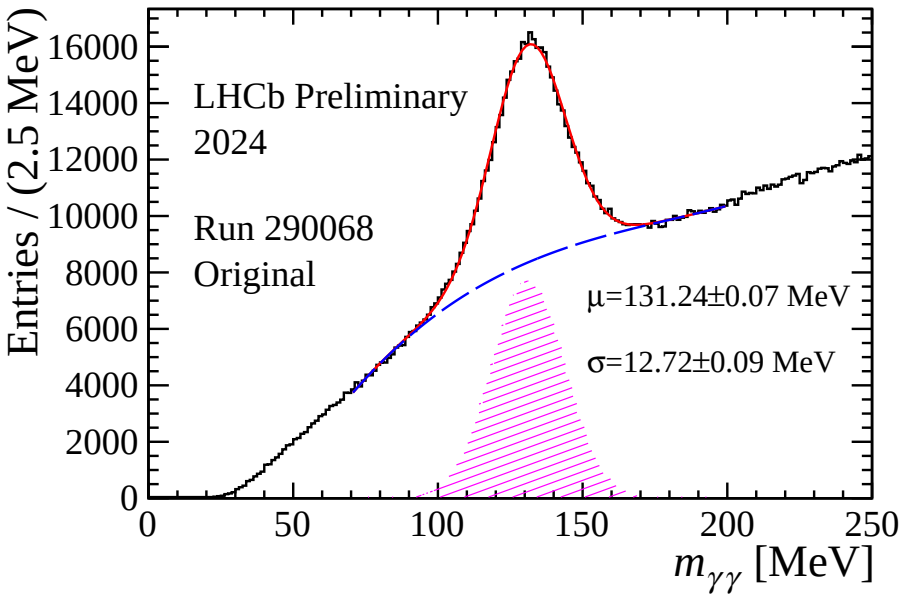
Hit Residual



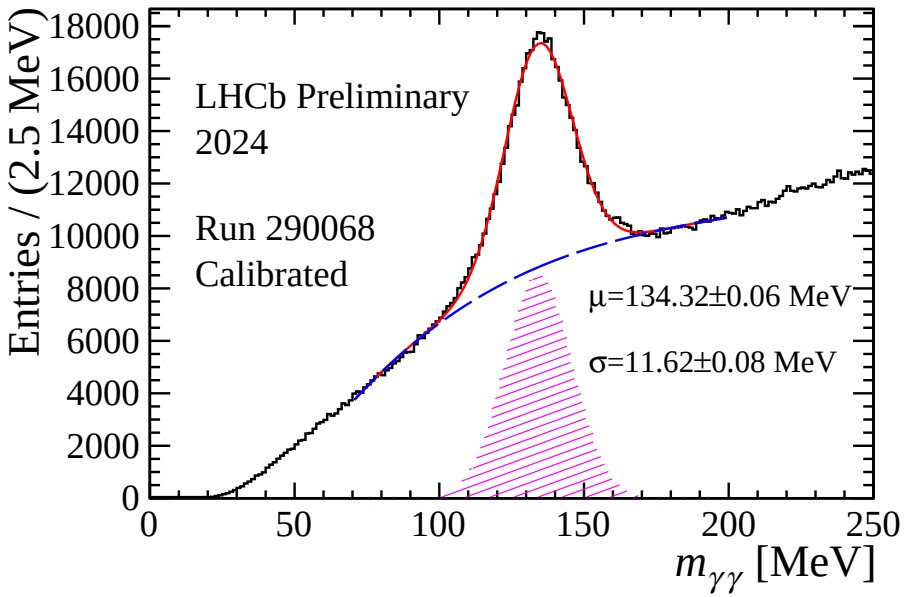
χ^2

π^0 Calibrations

- Calibration of ECAL brings $\pi^0 \to \gamma\gamma$ peak closer to PDG value 134.977 MeV.



Before



Calibrated

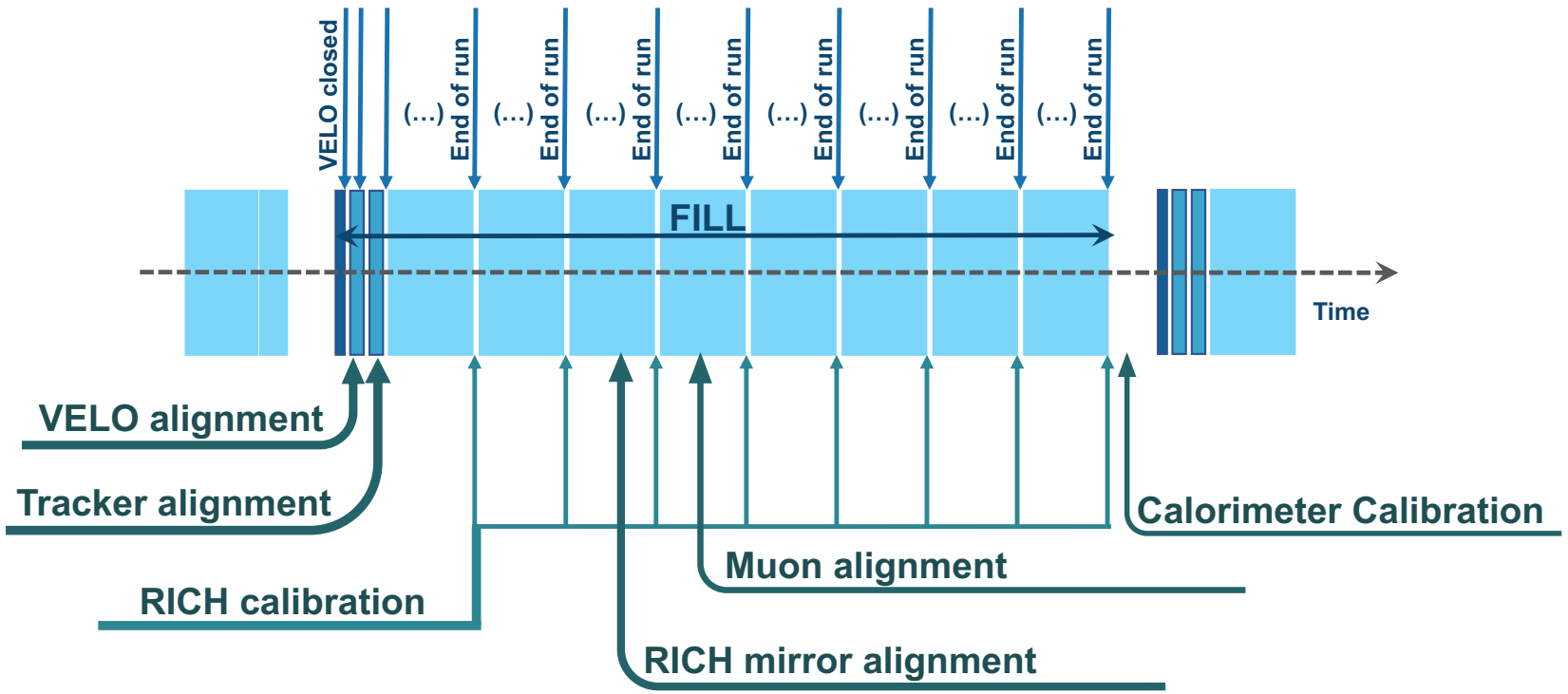
HLT 1 Cost Breakdown

- 20% cost savings with GPU farm.
- Figure from Computing and Software for Big Science 6 (2022) 1, 1.

Item	CPU-only	hybrid	Difference
Event Builder nodes	1000	1000	0
HLT1 network	275	25	250
HLT1 compute	450	125	325
Storage for 1 MHz output	575	575	0
Sub-total	2300	1725	575
Storage add. cost 2 MHz output	575	575	0
Total	2875	2300	575

Table 4: Indicative overall cost of the HLT1 implementations including contingency in units of the reference "Quanta" CPU server node used for the HLT during Run 2 datataking. Numbers have been rounded to reflect inevitable order(10%) fluctuations in real-world costs depending on the context of any given purchase.

Alignment and Calibration



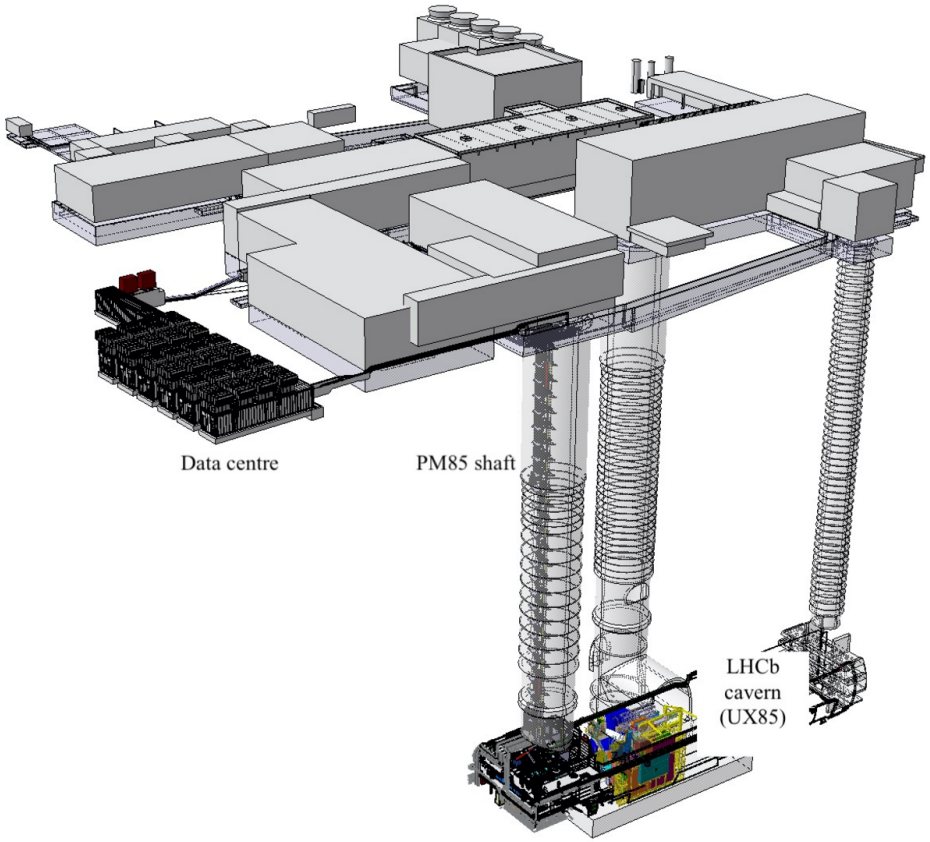
JINST 19 (2024) 05, P05065

- Different subdetectors require different statistics of tracks or physics signal to align and calibrate.
  - VELO closes every pp fill $\rightarrow$ per-fill alignment.
  - RICH is sensitive to temperature and pressure $\rightarrow$ per-run

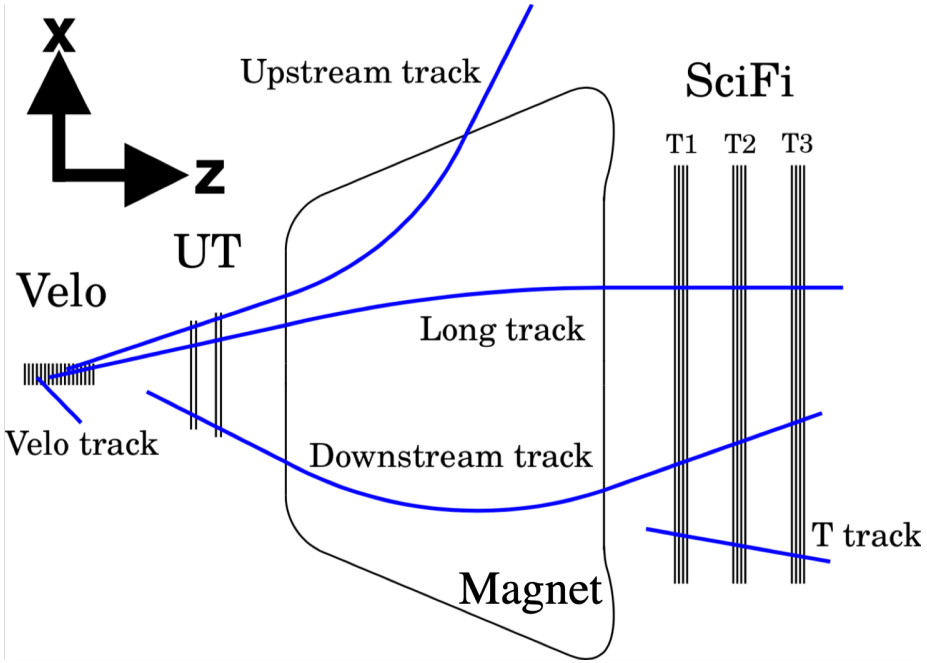
LHCb Data Center

JINST 19 (2024) 05, P05065

- LHCb has 250m optical fibres sending data from underground detector to custom data centre on the surface.
- Modular data centre hosts the HLT1 and HLT2 trigger farms.



LHCb Track Definitions



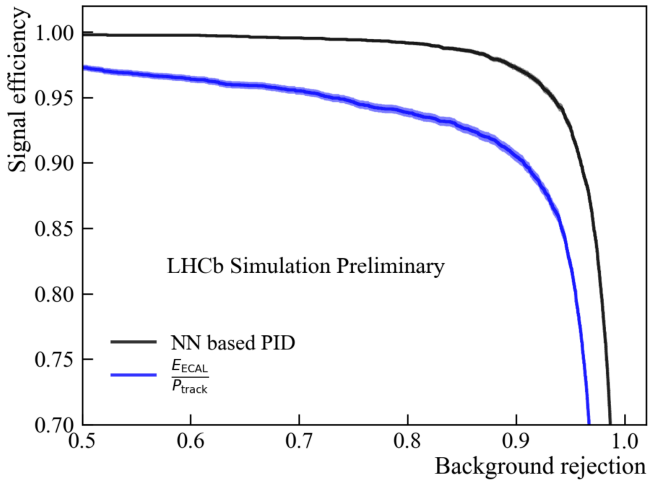
JINST 19 (2024) 05, P05065

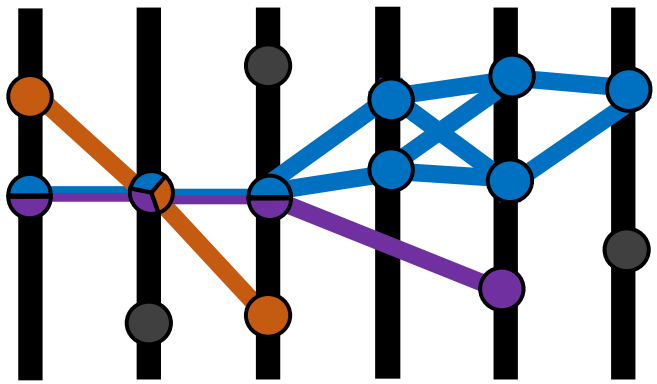
- Long tracks are the most important track types for physics analysis because they have vertex and momentum measurements.
- Downstream tracks have higher fake track rates but they are important for decays with neutral s hadrons, $K_s^0 \to \pi\pi$ and $\Lambda^0 \to p\pi$.

Machine Learning in RTA

- Lipschitz NN. arxiv:2112.00038
  - Small and fast.
  - Robust and monotonic.
  - Used in PID and trigger selections.

LHCB-FIGURE-2024-003

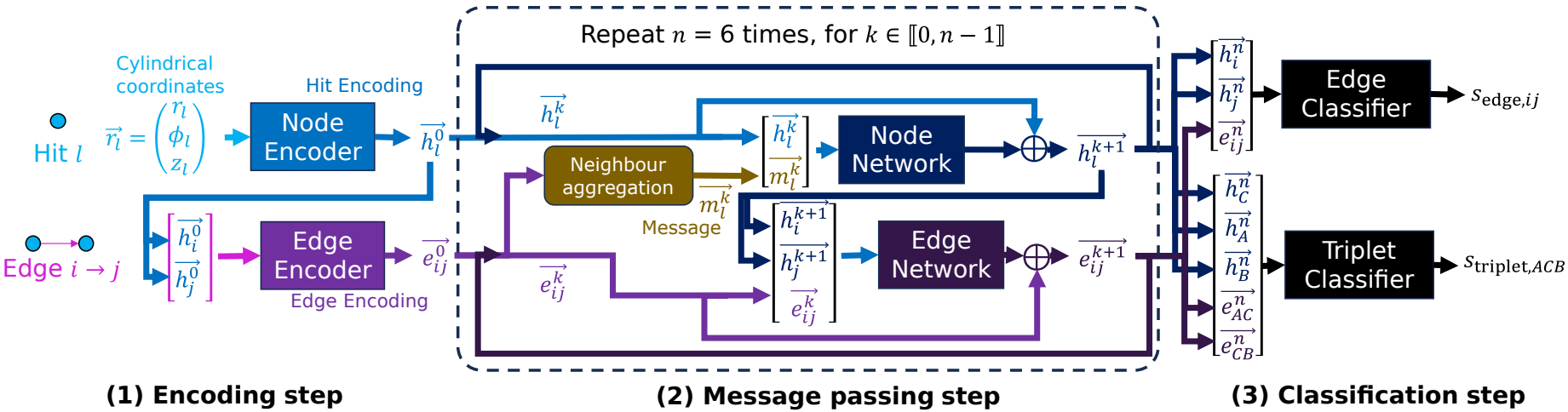




arXiv:2407.12119

- Use GNN to reconstruct tracks in the VELO.
- Paper in review, should on arXiv in a few weeks.

Machine Learning in RTA



arXiv:2407.12119

- GNN for VELO tracking:
  - Hits are nodes.
  - Connections between hits are edges.
  - Edge connections $\rightarrow$ triplet.