Real-time data selection on GPUs for the LHCb experiment

Dorothea vom Bruch

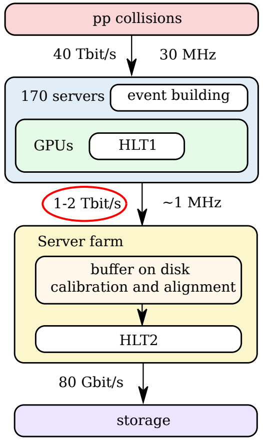
Aix Marseille Univ, CNRS/IN2P3, CPPM

Niels Bohr lunch seminar

November 20th 2020

Outline

- Introduction to LHCb and its upgrades
- Real-time data selection at LHCb
- Introducing Graphics Processing Units (GPUs)



- How GPUs will be used in real-time data selection at LHCb from 2022 onwards

LHCb and its upgrades

Search for new physics

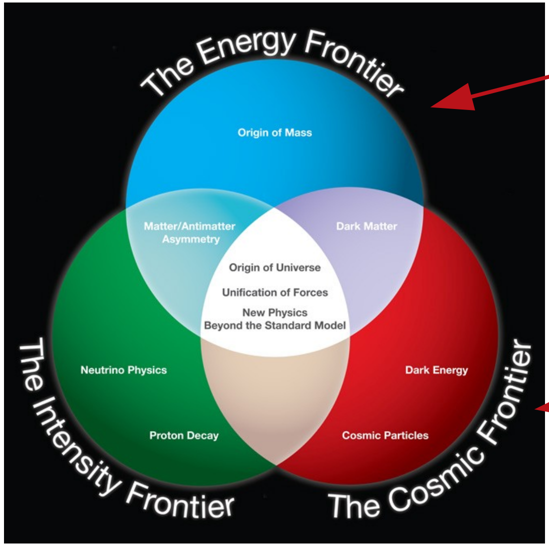
Option: Indirect searches

- Precision measurements of precisely calculated observables
- Null-tests: Search for forbidden processes
- Deviations: Precise measurement of precisely calculated observable

Option: Direct searches

- Directly detect new particles
- Either at dedicated experiments (WIMPs, Axions etc.)
- Or at ever increasing energy scales

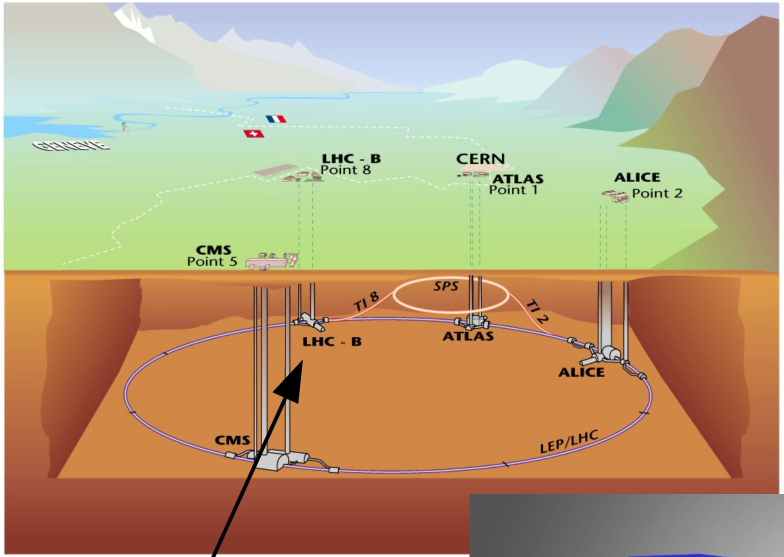
For example at LHCb



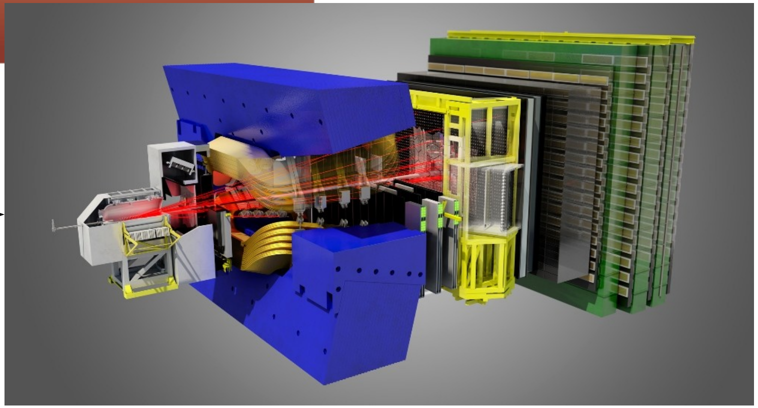
Astroparticle experiments

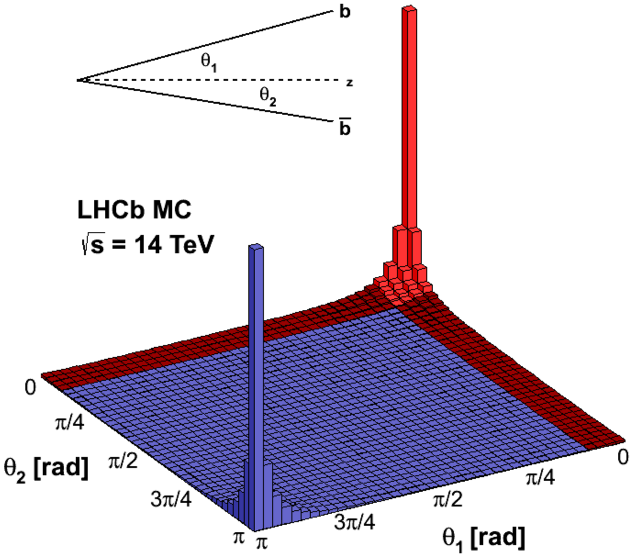
LHCb @ the LHC

LHC @ CERN



General purpose detector in the forward region specialized in beauty and charm hadrons

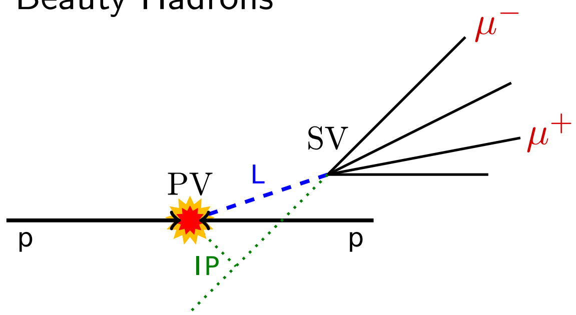




C. Elsässer, bb production angle plots

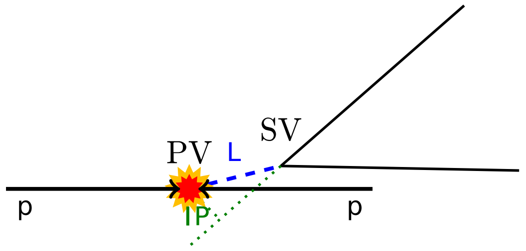
Beauty and charm decays

Beauty Hadrons



- $B^{\pm/0}$ mass ~5.3 GeV
  → Daughter p_T $\mathcal{O}$(1 GeV)
- τ ~1.6 ps → flight distance ~1cm
- Detached muons from B→J/ΨX, J/Ψ → $\mu^+\mu^-$
- Displaced tracks with high p_T

Charm Hadrons

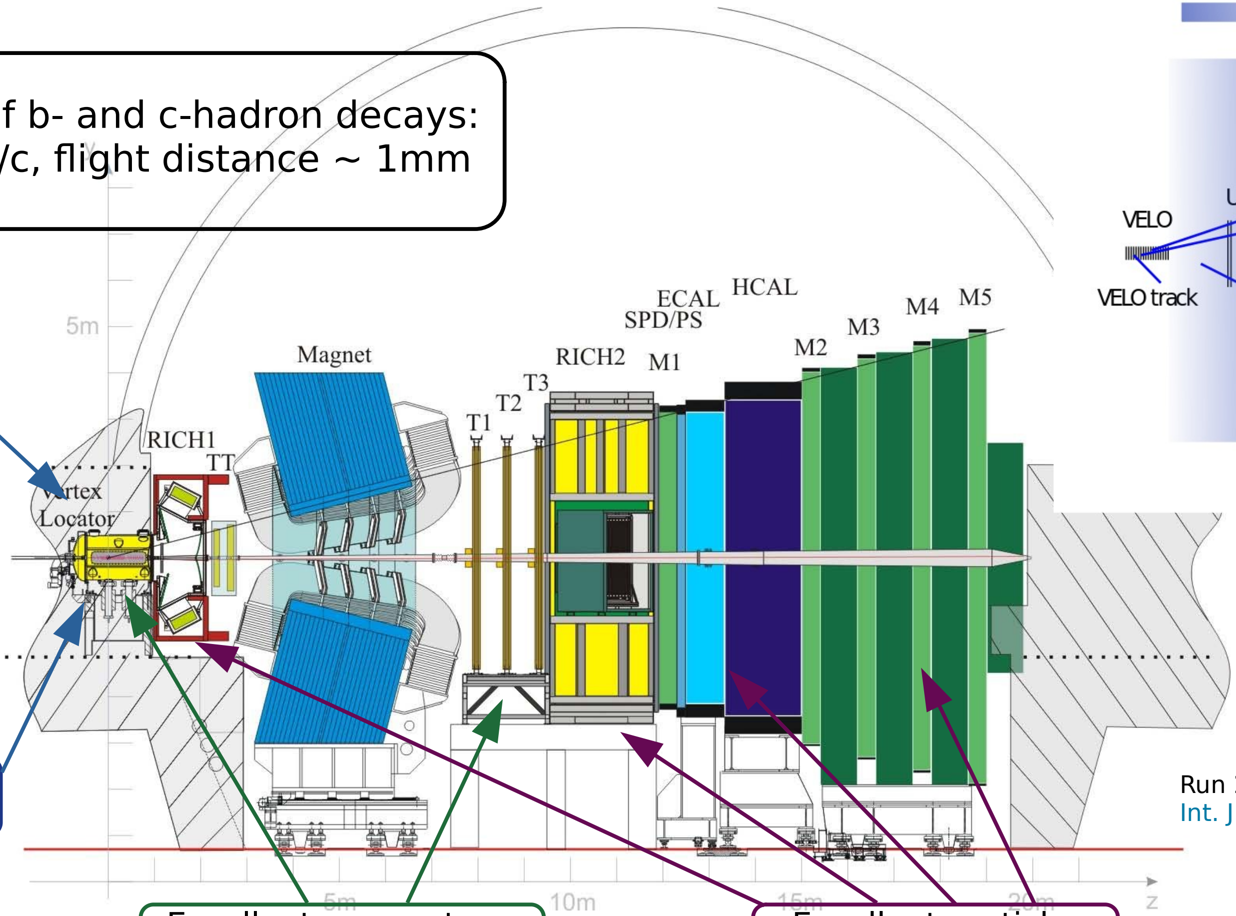


- $D^{\pm/0}$ mass ~1.9 GeV
  → Daughter p_T $\mathcal{O}$(700 MeV)
- τ ~0.4 ps → flight distance ~4mm
- Also produced from B decays

PV: Primary vertex
SV: Secondary vertex
IP: Impact parameter (distance between point of closest approach of a track and a PV)

LHCb detector, 2011 - 2018

Daughters of b- and c-hadron decays:
$p_T \sim 1$ GeV/c, flight distance ~ 1mm

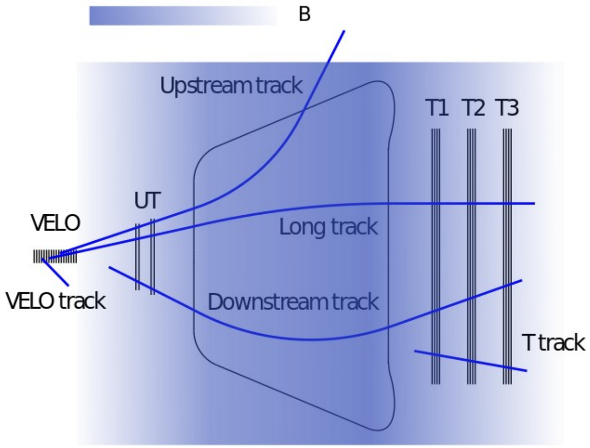


Precise vertex measurements

Excellent decay time resolution

Excellent momentum resolution

Excellent particle identification

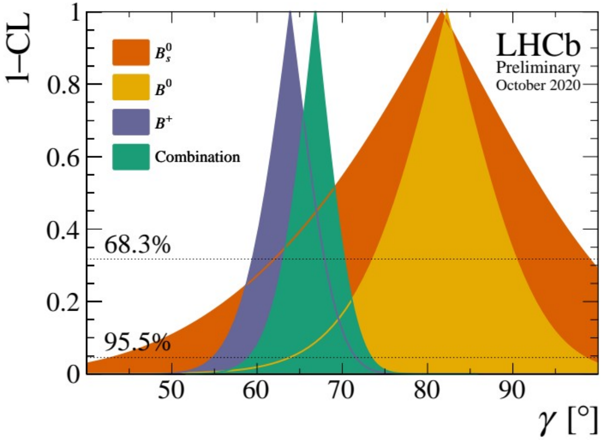


Run 2 detector performance:
Int. J. Mod. Phys. A30 (2015) 1530022

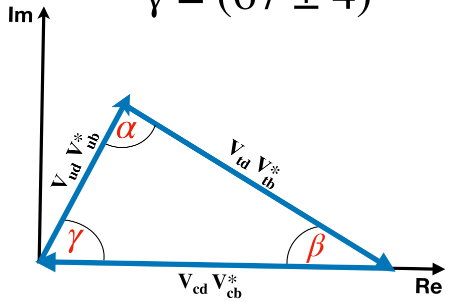
Highlights from Runs 1 & 2

Constraining CKM angles

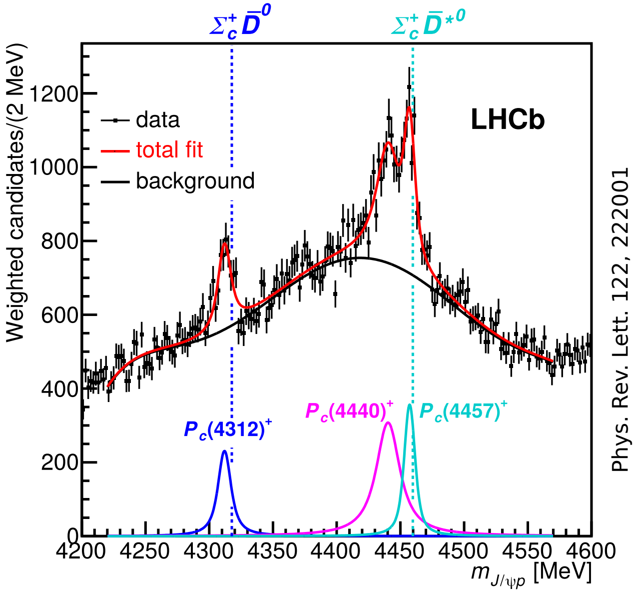
LHCb-CONF-2020-003



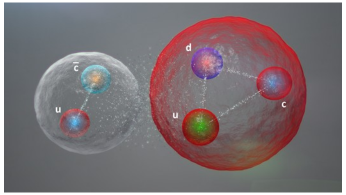
$\gamma = (67 \pm 4)^\circ$



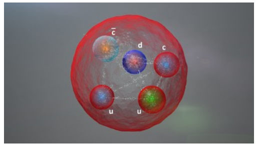
Pentaquarks



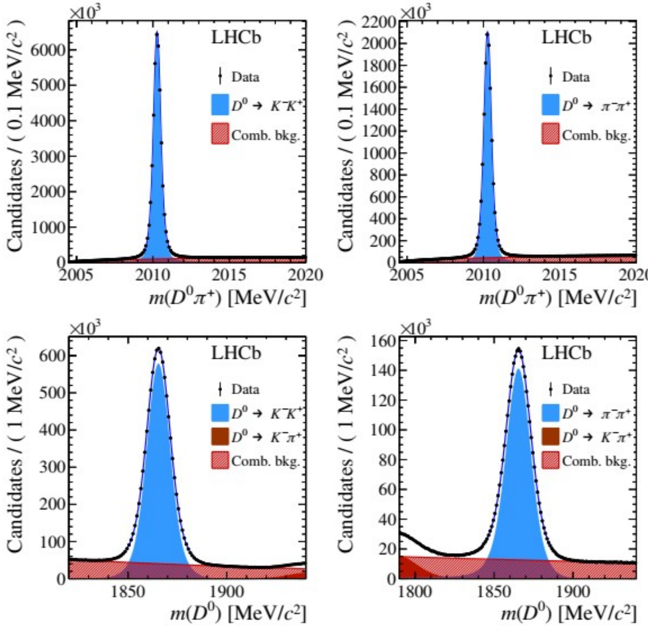
Phys. Rev. Lett. 122, 222001



or



CP violation in charm decays



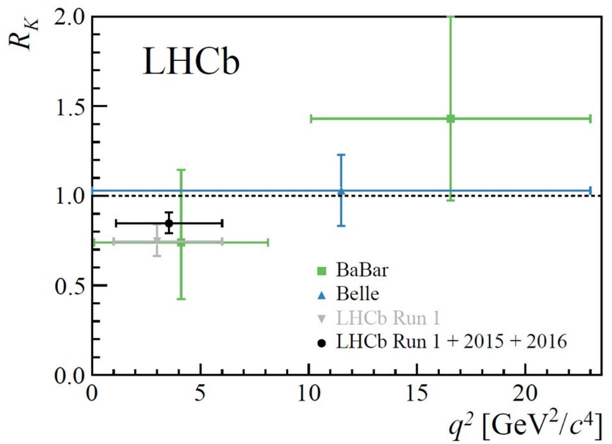
Phys. Rev. Lett. 122, 211803 (2019)

$\Delta A_{CP} = (-15.4 \pm 2.0) \times 10^{-4}$

Highlights from Runs 1 & 2

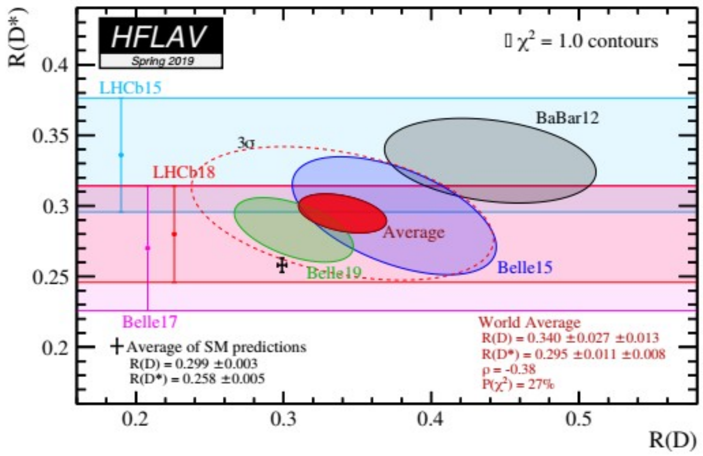
Lepton flavor universality

$b \to s l^+ l^-$



T. Humair, Moriond 2019

$b \to c l \nu$



$\mathcal{R}(K^{(*)}) = \mathcal{B}(B \to K^{(*)}\mu^+\mu^-) / \mathcal{B}(B \to K^{(*)}e^+e^-)$

$\mathcal{R}(D^{(*)}) = \mathcal{B}(B \to D^{(*)}\tau\nu_\tau) / \mathcal{B}(B \to D^{(*)}\mu(e)\nu_{\mu(e)})$

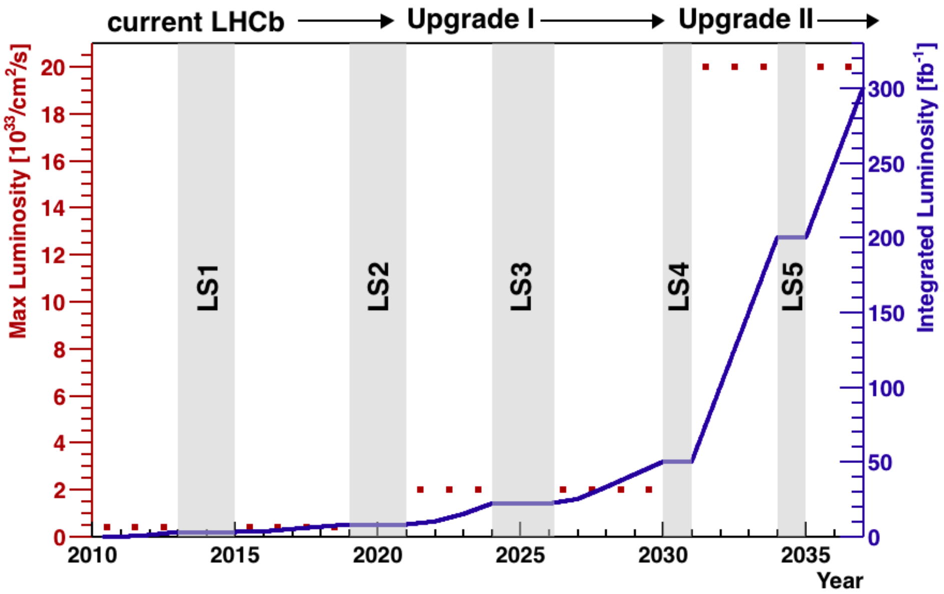
R(D) and R(D*) compatile with the SM at the 3.1 σ level

R(K) and R(K*) are compatible with the SM at 2.5 σ and 2.1-2.5 σ respectively

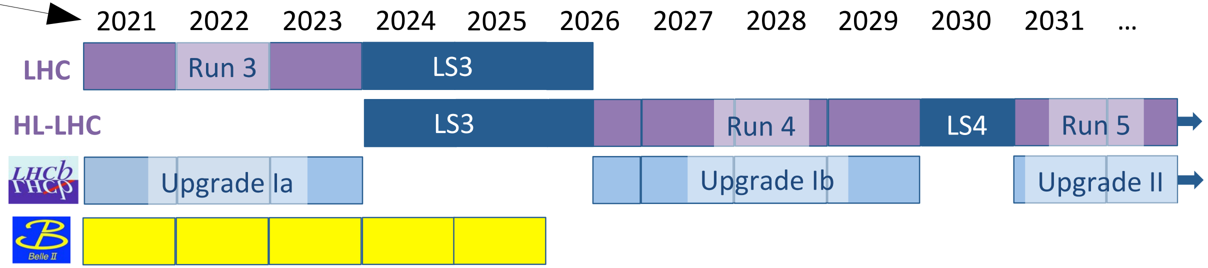
LHCb Upgrades

Push the intensity frontier

- Study more pp-bunch collisions per second
- Detectors with at least the same precision
- Significantly reduce uncertainties

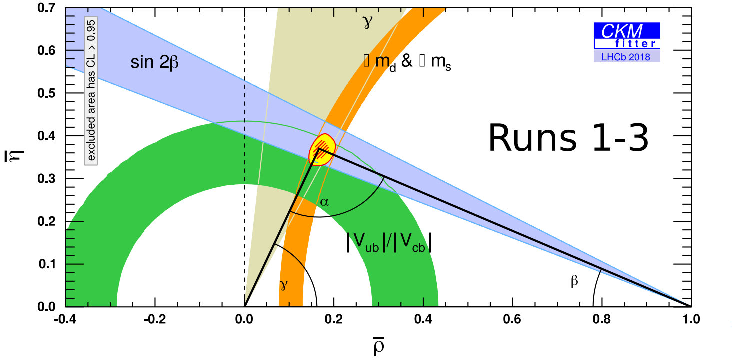


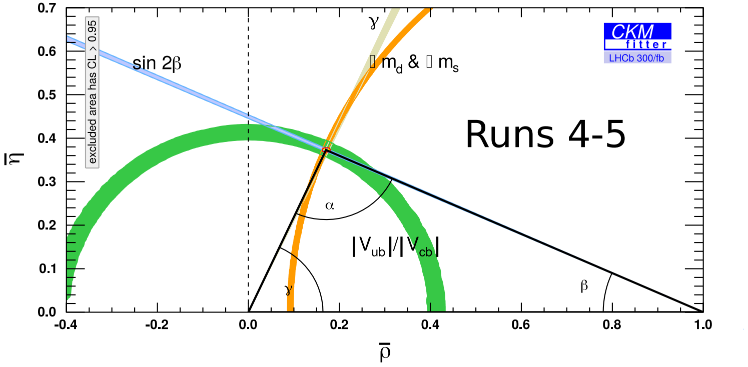
Delayed to February 2022

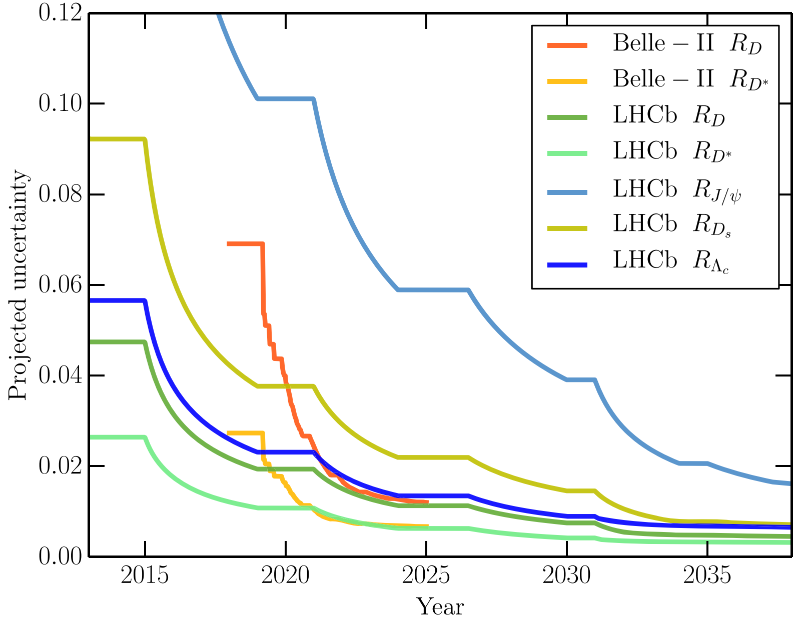


Prospects for Run 3 and beyond

https://arxiv.org/pdf/1808.08865.pdf



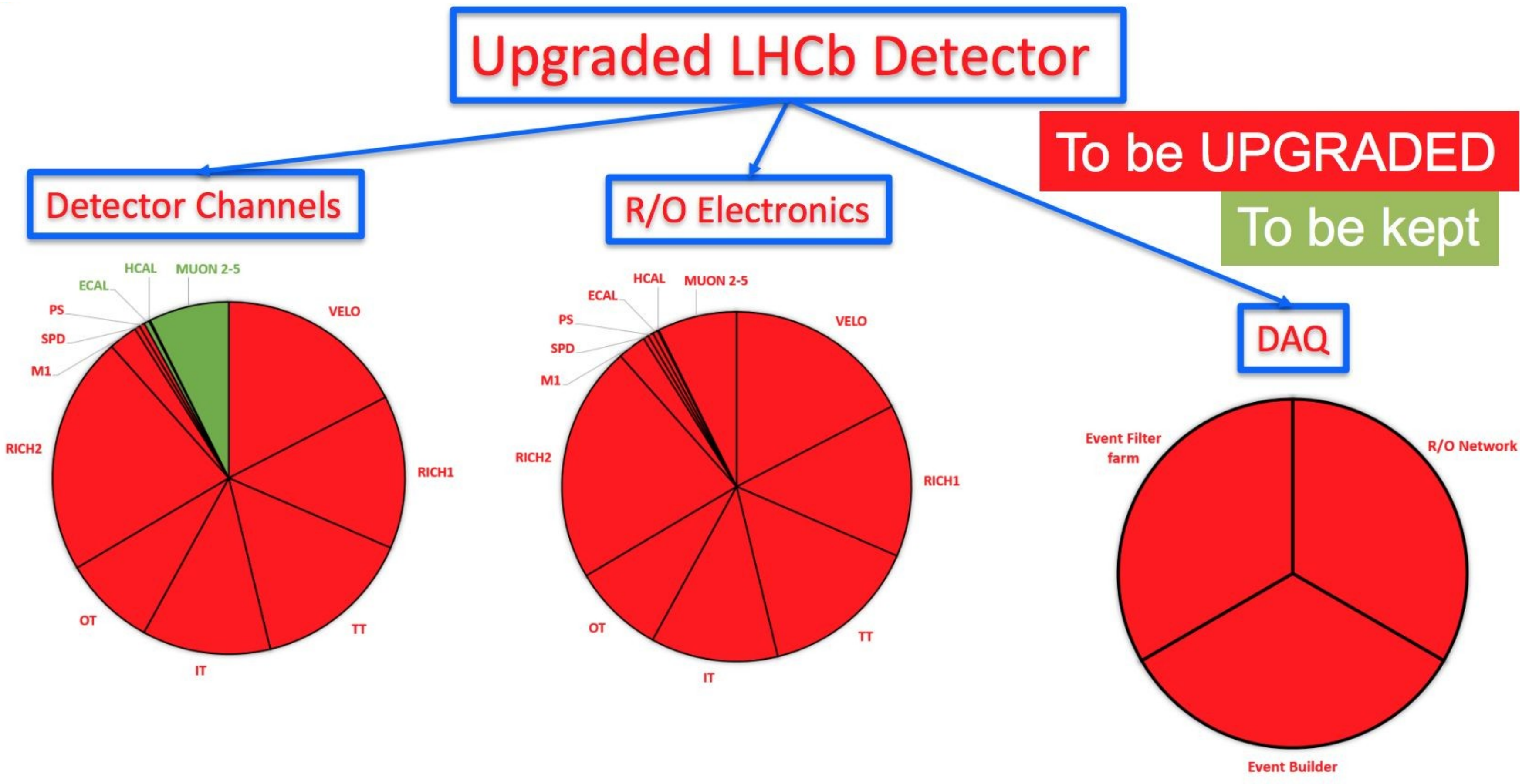




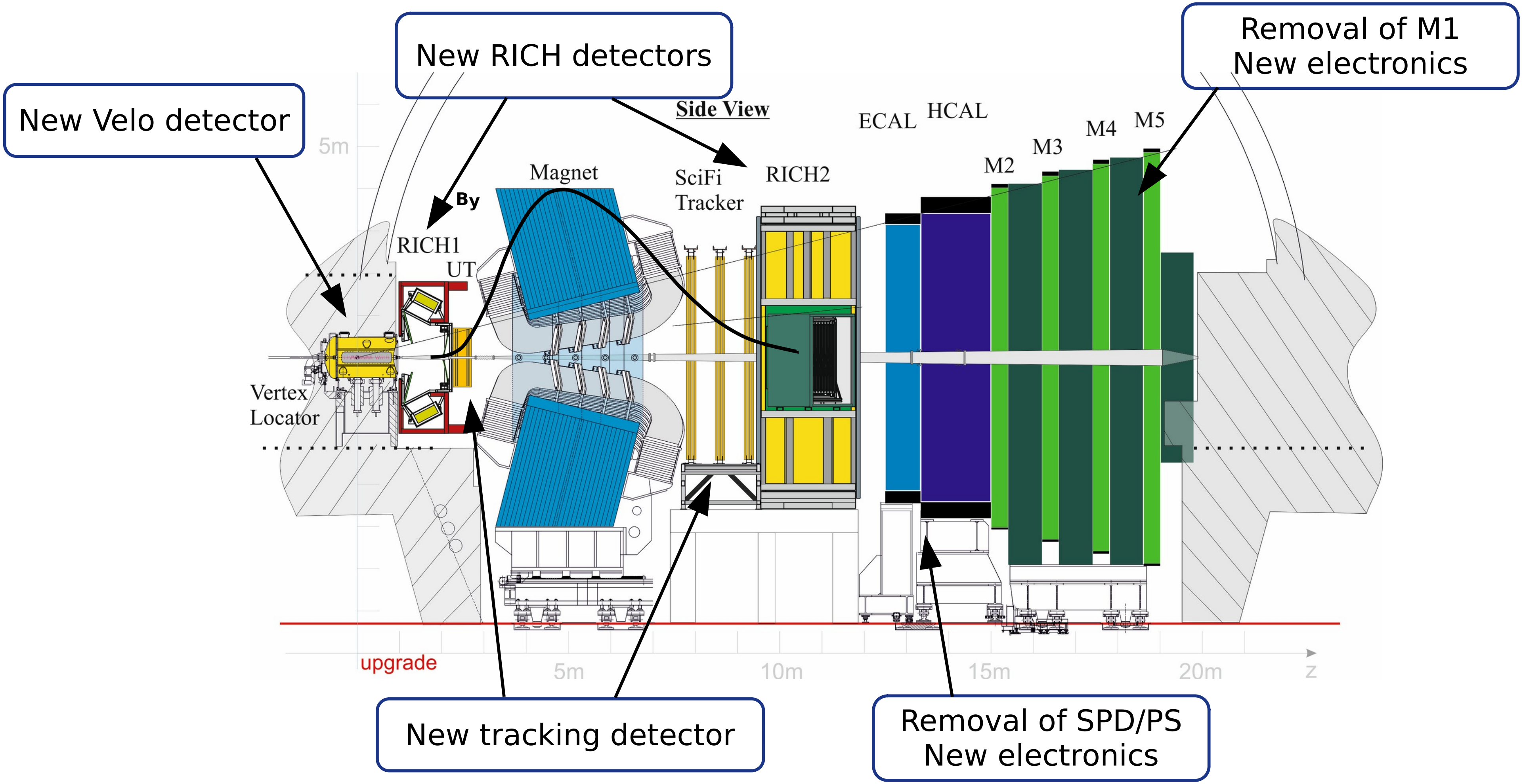
Run 3 and beyond will shed light on the flavor anomalies observed today

Precise and efficient data selection key to fully the exploiting physics potential

LHCb Upgrade I

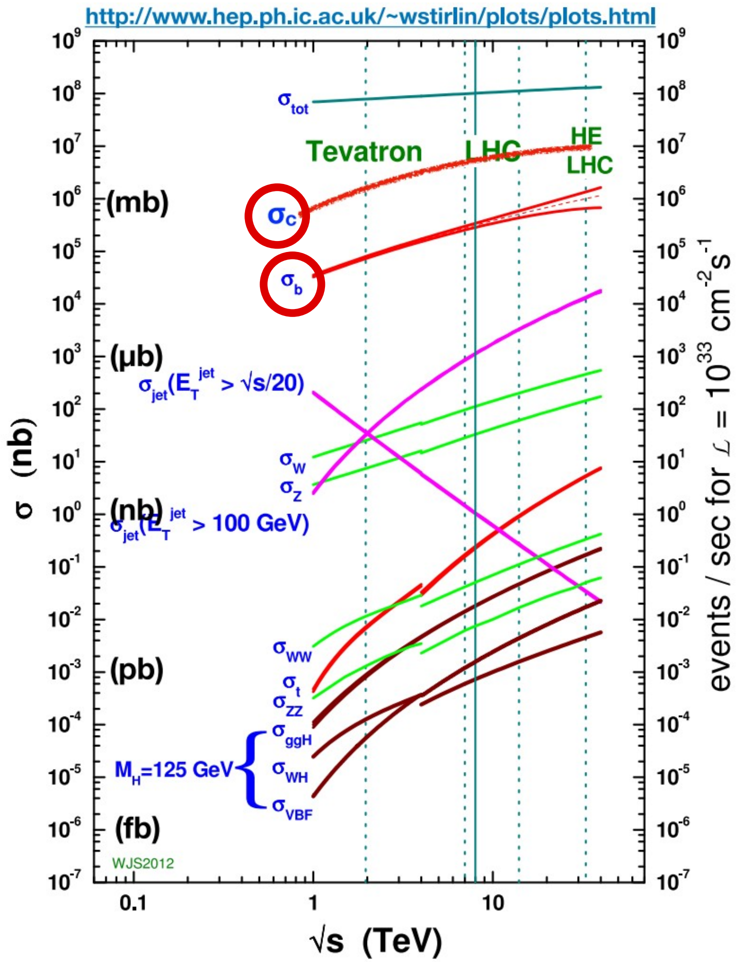


LHCb detector in Run 3



Real-time data selection at LHCb

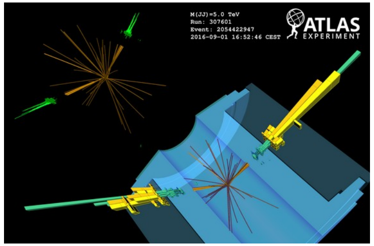
The MHz signal era



Run 3: Luminosity of 2×10^{33} $cm^{-2}s^{-1}$, $\sqrt{s}$ = 14 TeV

General purpose LHC experiments:

- Mainly direct searches
- Local characteristic signatures
- Signal rates up to ~100 kHz

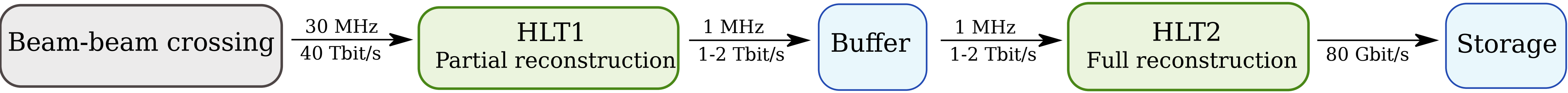


LHCb:

- Intensity frontier
- No “simple” local criteria for selection
- Signal rates up to ~MHz
- Access as much information about the collision as early as possible
- Read out the full detector

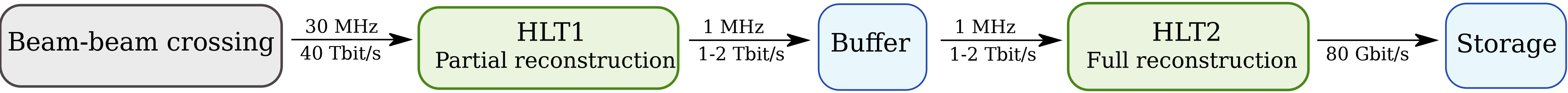
Change in real-time data selection paradigm

Data selection only in software



- **High Level Trigger 1 (HLT1):**
  - Full charged particle track reconstruction
  - Few inclusive single and two-track selections
- **High Level Trigger 2 (HLT2):**
  - Real-time aligned and calibrated detector
  - Offline-quality track reconstruction
  - Particle identification
  - Full track fit

Data selection only in software



- **High Level Trigger 1 (HLT1):**
  - Full charged particle track reconstruction
  - Few inclusive single and two-track selections
- **High Level Trigger 2 (HLT2):**
  - Real-time aligned and calibrated detector
  - Offline-quality track reconstruction
  - Particle identification
  - Full track fit

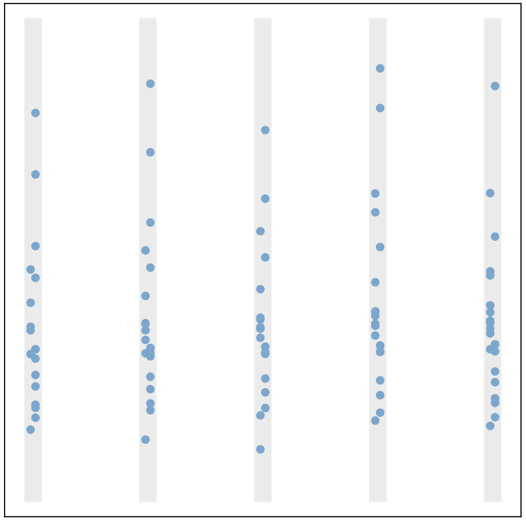
Comparison to Run II trigger

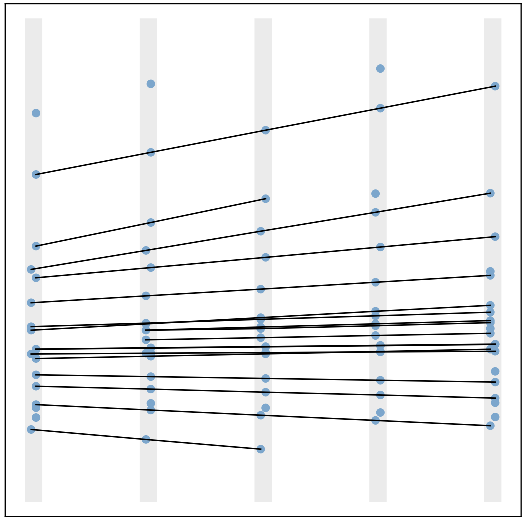
- 5 x higher pileup
- 30 x higher rate into HLT1
- Disk buffer reduces from O(weeks) → O(days)
- Up to 10 x efficiency improvement for some physics channels

Huge computing challenge

Track reconstruction @ 30 MHz

- Connect the dots to go from measurements to particle trajectories
- Many possible connections → huge combinatorics
- Do this for three sub-detectors, 30 million times per second



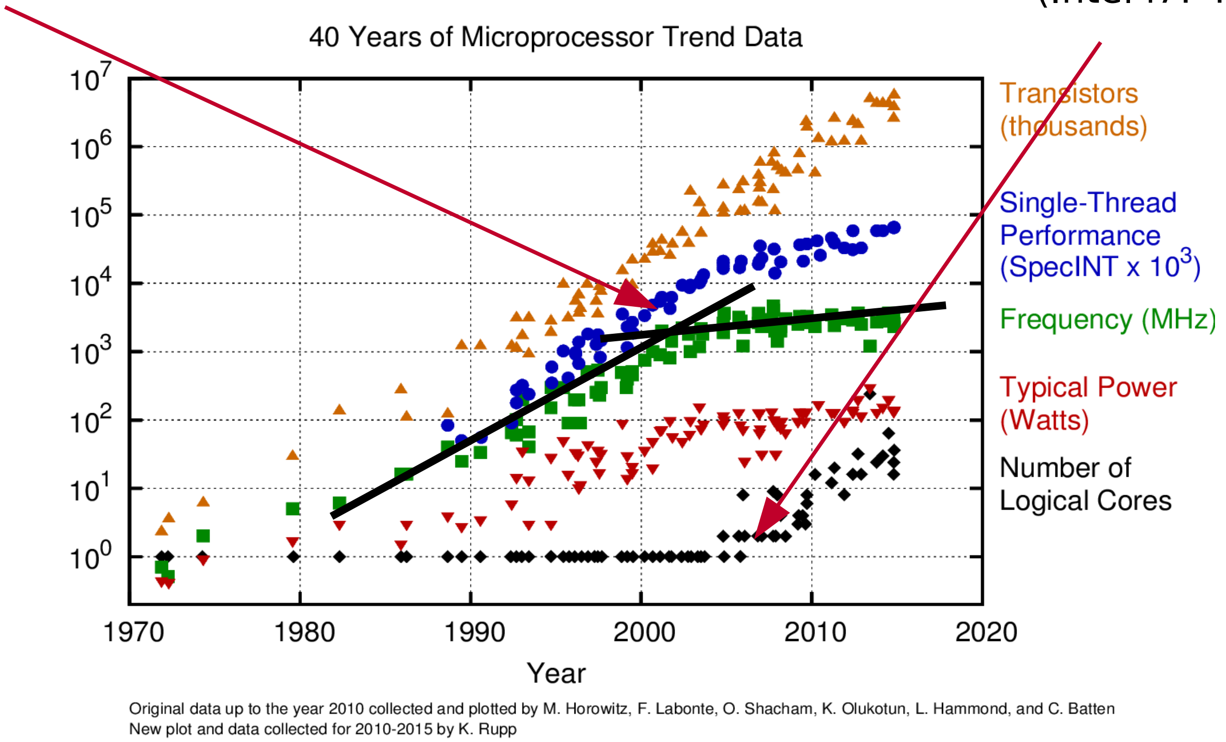


Introducing Graphics Processing Units (GPUs)

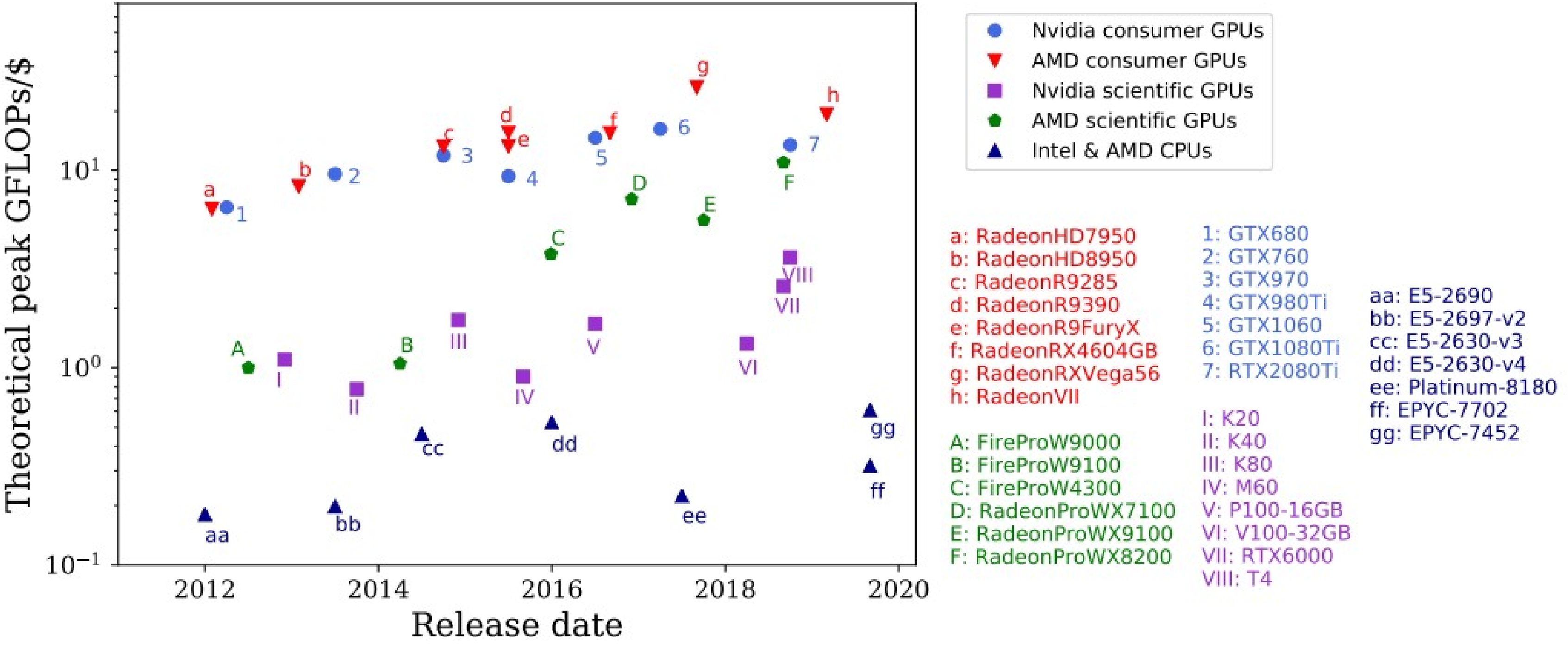
Moore’s law today

Clock speed stopped increasing due to heat limit

Multiple core processors emerge (Intel i7: 4 cores)



Theoretical FLOPs/$: GPUs & CPUs

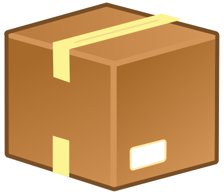


JINST 15 C06010 (2020)

Why the GPU computing trend?

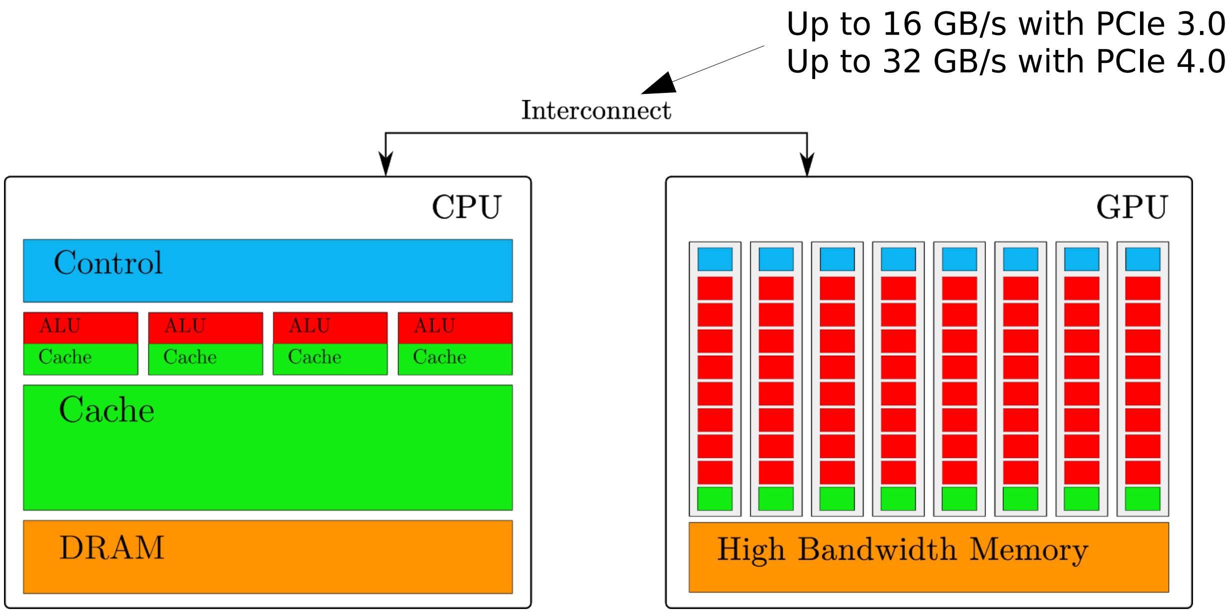
Best theoretical FLOPs/$

Power efficient



Many FLOPs in one device
→ compact system possible

GPU architecture design



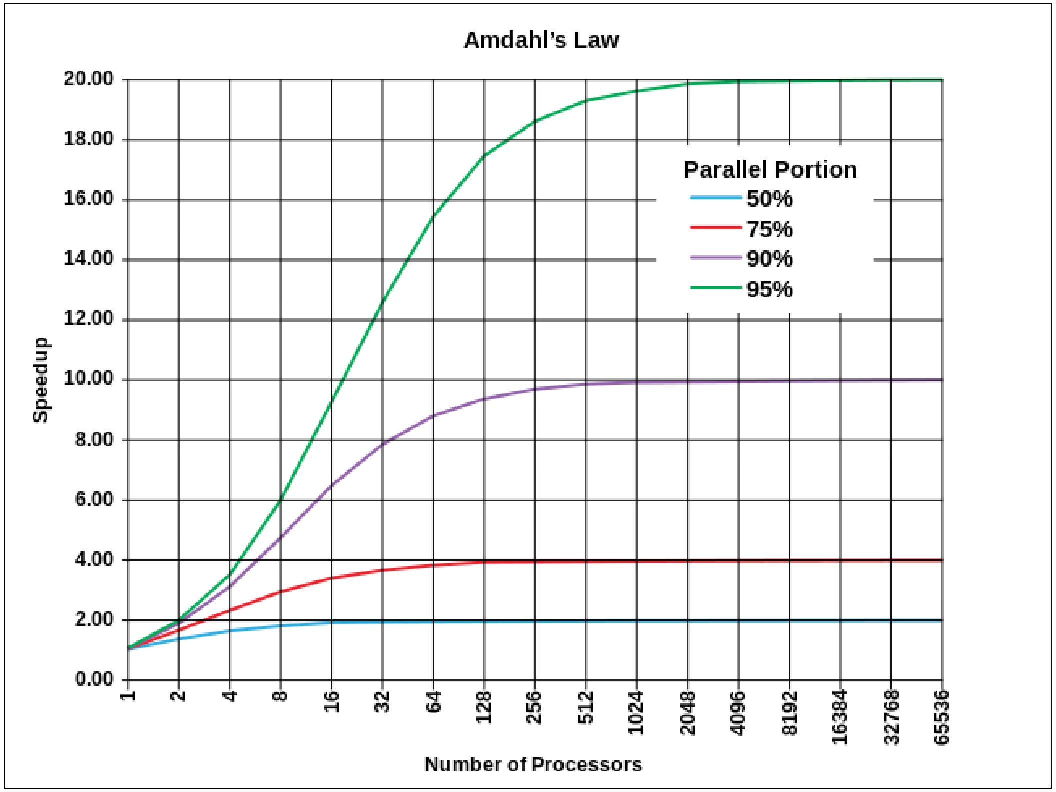
- Low core count / powerful ALU
- Complex control unit
- Large chaches

→ **Latency optimized**

- High core count
- No complex control unit
- Small chaches

→ **Throughput optimized**

When to go parallel? → Amdahl’s law

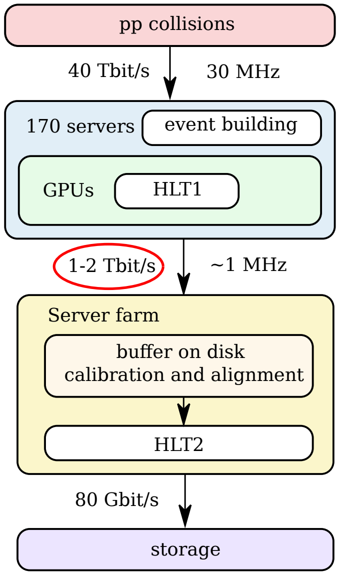


Speedup in latency = 1 / (S + P/N)

- S: sequential part of program
- P: parallel part of program
- N: number of processors

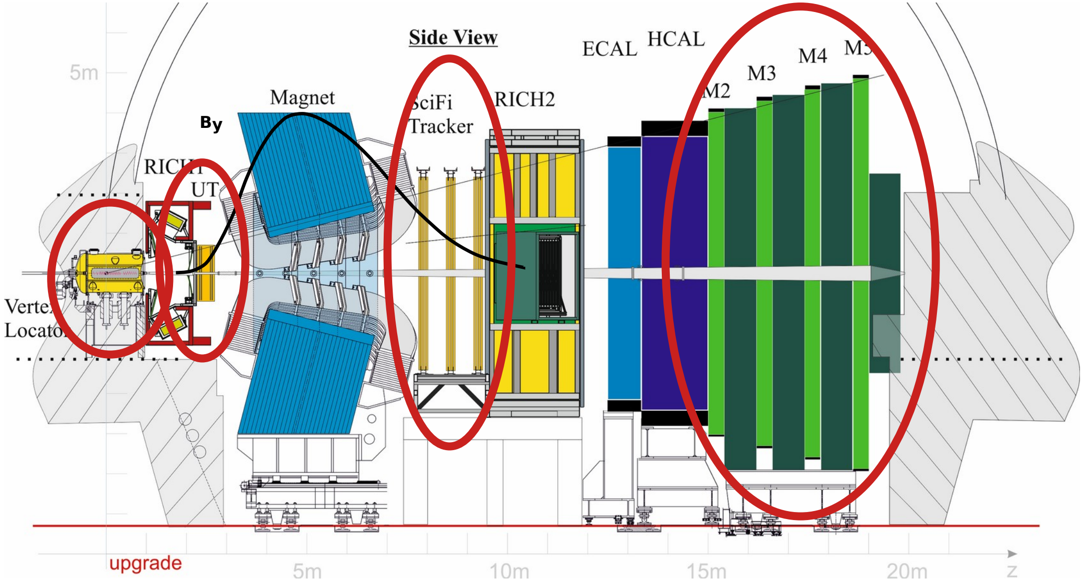
- Parallel part: identical, but independent work
- Consider how much of the problem can actually be parallelized!

GPUs in LHCb’s High Level Trigger 1 (HLT1)



LHCb HLT1 elements

- Decode binary payload of four sub-detectors
- Reconstruct charged particle trajectories
- Identify muons
- Reconstruct primary and secondary decay vertices
- Select pp-bunch collisions based on
  - Single-track properties
  - Secondary vertex properties



- Manageable amount of algorithms with highly parallelizable tasks
- Raw event size O(100) kB
- **Can copy full event information to GPU and implement & optimize all HLT1 algorithms to run efficiently on a GPU**

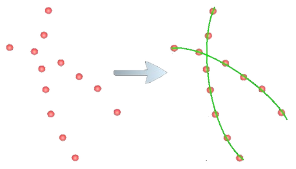
Common parallelization techniques

Raw data decoding

- Transform binary payload from subdetector raw banks into collections of hits (x,y,z) in LHCb coordinate system
- Parallelize over all subdetectors and readout units

Track reconstruction

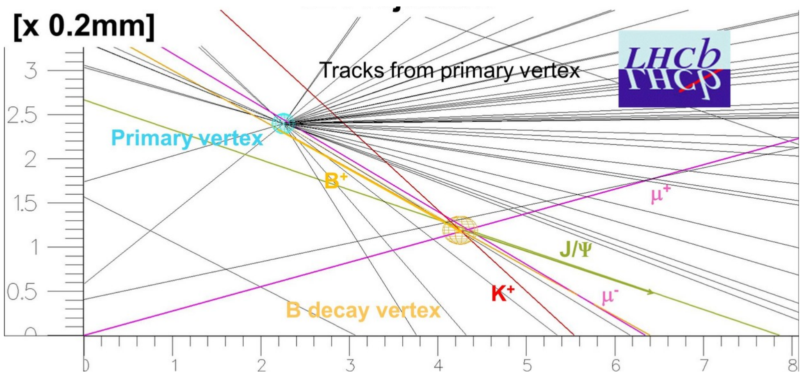
- Consists of two steps:
  - Pattern recognition: Which hits belong to which track?
  - Track fitting: Done for every track
- Parallelize over combinations of hits and tracks



f(x) = … +/- …

Vertex finding

- Reconstruct primary and secondary vertices
- Parallelize across combinations of tracks and vertex seeds



How does the HLT1 map to GPUs?

Characteristics of LHCb HLT1	**Characteristics of GPUs**
Intrinsically parallel problem: - Run events in parallel - Reconstruct tracks in parallel	Good for - Data-intensive parallelizable applications - High throughput applications

How does the HLT1 map to GPUs?

Characteristics of LHCb HLT1	**Characteristics of GPUs**
Intrinsically parallel problem: - Run events in parallel - Reconstruct tracks in parallel	Good for - Data-intensive parallelizable applications - High throughput applications
Huge compute load	Many TFLOPS

How does the HLT1 map to GPUs?

Characteristics of LHCb HLT1	**Characteristics of GPUs**
Intrinsically parallel problem: - Run events in parallel - Reconstruct tracks in parallel	Good for - Data-intensive parallelizable applications - High throughput applications
Huge compute load	Many TFLOPS
Full data stream from all detectors is read out → no stringent latency requirements	GPUs have higher latency than CPUs, not as predictable as FPGAs

How does the HLT1 map to GPUs?

Characteristics of LHCb HLT1	**Characteristics of GPUs**
Intrinsically parallel problem: - Run events in parallel - Reconstruct tracks in parallel	Good for - Data-intensive parallelizable applications - High throughput applications
Huge compute load	Many TFLOPS
Full data stream from all detectors is read out → no stringent latency requirements	GPUs have higher latency than CPUs, not as predictable as FPGAs
Small raw event data (~100 kB)	Connection via PCIe → limited I/O bandwidth

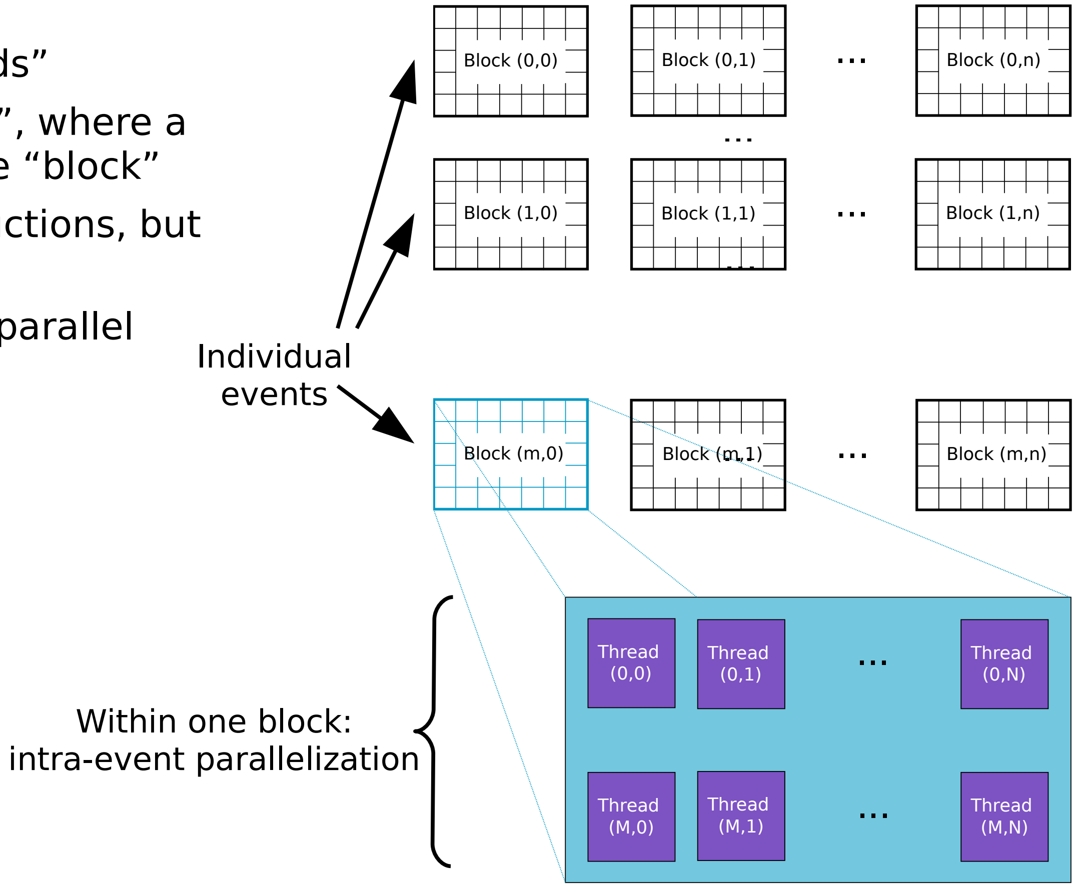
How does the HLT1 map to GPUs?

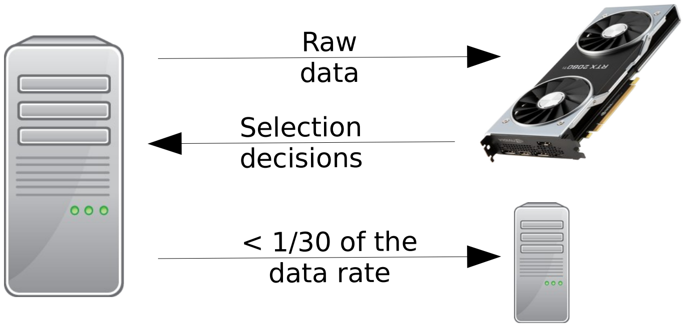
Characteristics of LHCb HLT1	**Characteristics of GPUs**
Intrinsically parallel problem: - Run events in parallel - Reconstruct tracks in parallel	Good for - Data-intensive parallelizable applications - High throughput applications
Huge compute load	Many TFLOPS
Full data stream from all detectors is read out → no stringent latency requirements	GPUs have higher latency than CPUs, not as predictable as FPGAs
Small raw event data (~100 kB)	Connection via PCIe → limited I/O bandwidth
Small event raw data (~100 kB)	Thousands of events fit into O(10) GB of memory

Perfect fit!

HLT1 on GPUs

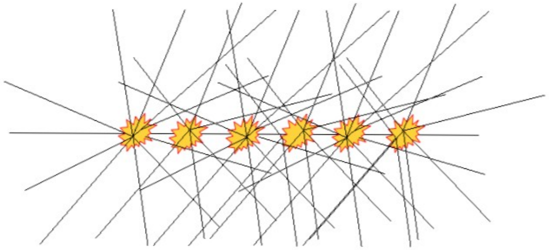
- GPU code is executed on many “threads”
- These threads are organized in a “grid”, where a fixed set of threads is grouped into one “block”
- Each thread processes the same instructions, but on different data
- Thousands of events are processed in parallel
- In addition: intra-event parallelization
- Only single precision is used

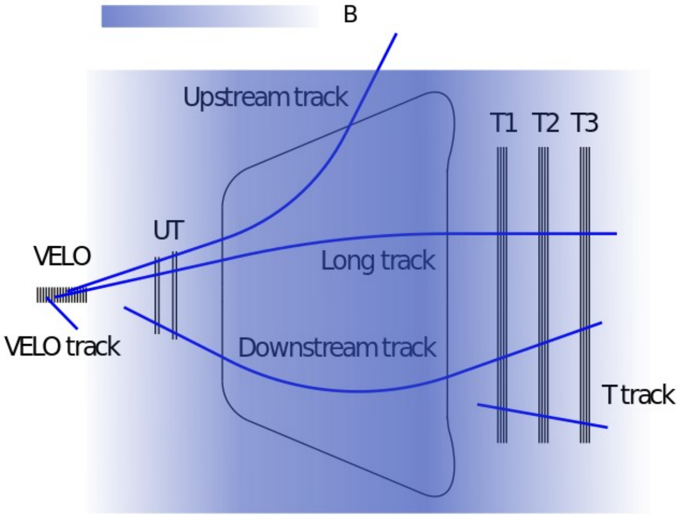




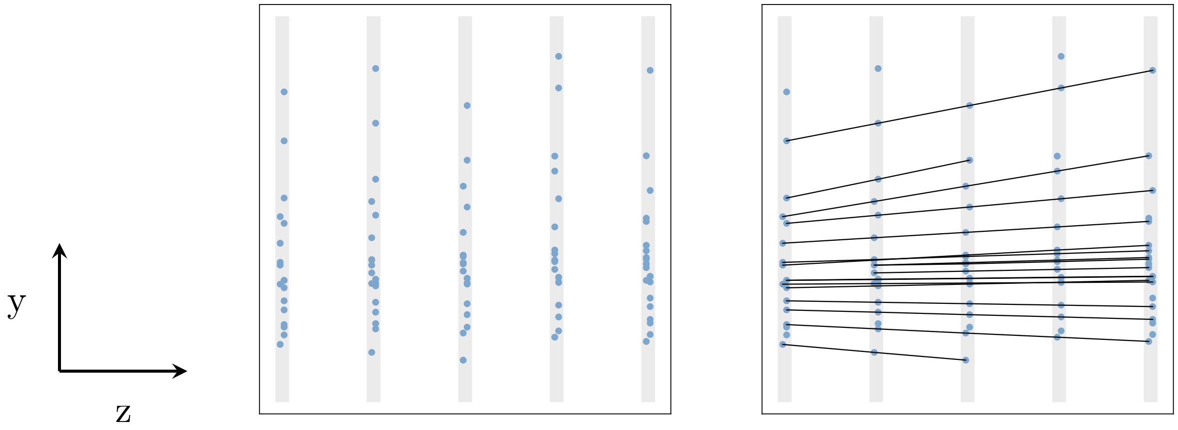
LHCb: Characteristics for pattern recognition

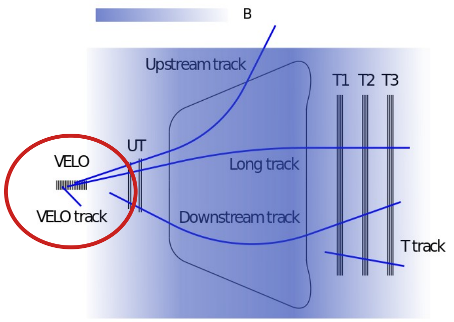
- Average pile up of 6
- Few hundred - few thousand hits in subdetectors
- Tens to hundreds of tracks in subdetectors
- Velo tracks are input for:
    - Primary vertex finding
    - Track forwarding to other detectors
- Mainly straight line tracks
- Large bend between UT and SciFi detectors
- Need curvature in magnetic field for good extrapolation to next subdetector
- Most tracks have $p_T < 2$ GeV/c





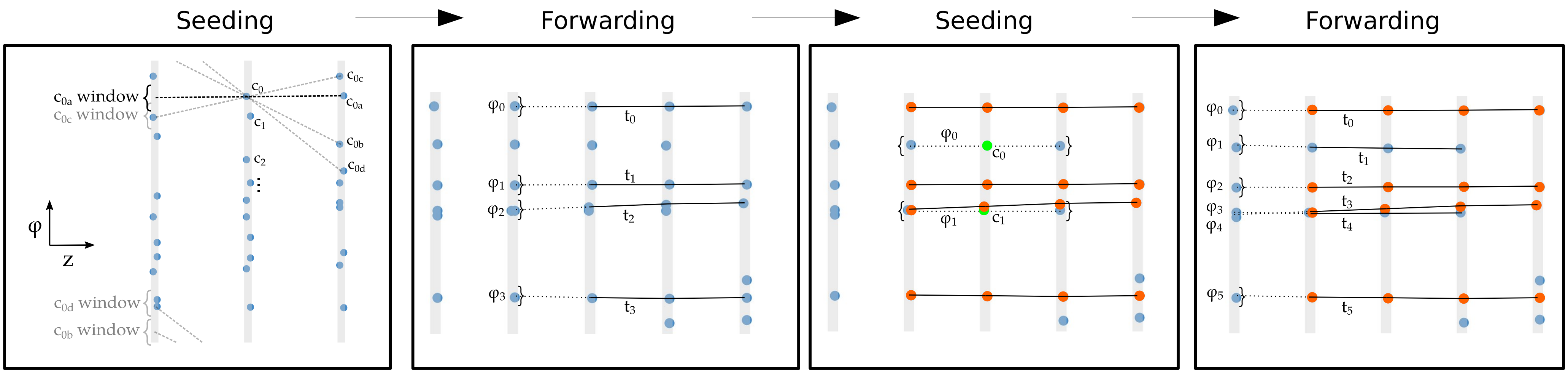
Velo track reconstruction





- No magnetic field in the Velo detector
- → straight line tracks
- Tracks from origin traverse detector in line of constant phi

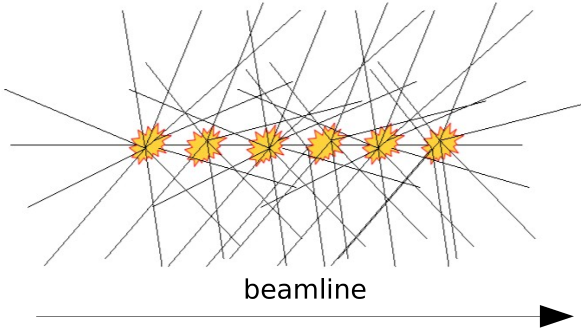
Velo track reconstruction on GPUs

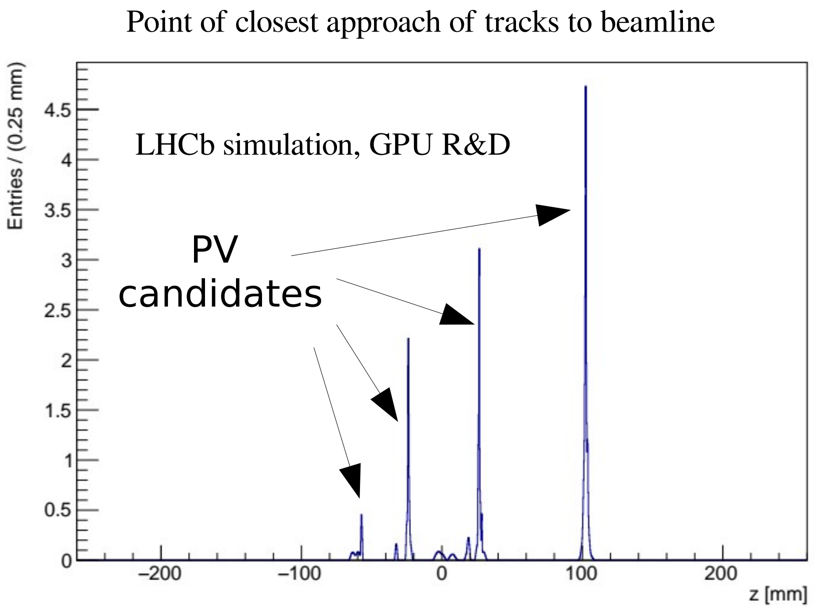


- Build “triplets” of three hits on consecutive layers → parallelization
- Choose them based on alignment in phi
- Hits sorted by phi → memory accesses as contiguous as possible
- Extend triplets to next layer → parallelization

D. Campora, N. Neufeld, A. Riscos Núñez: “A fast local algorithm for track reconstruction on parallel architectures”, IPDPSW 2019

Primary vertex reconstruction

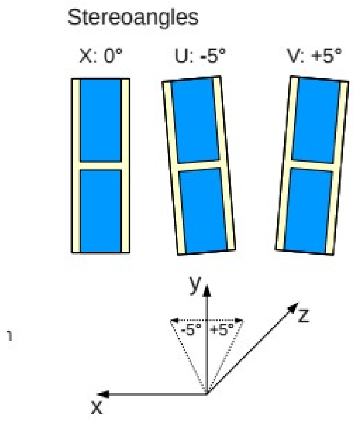


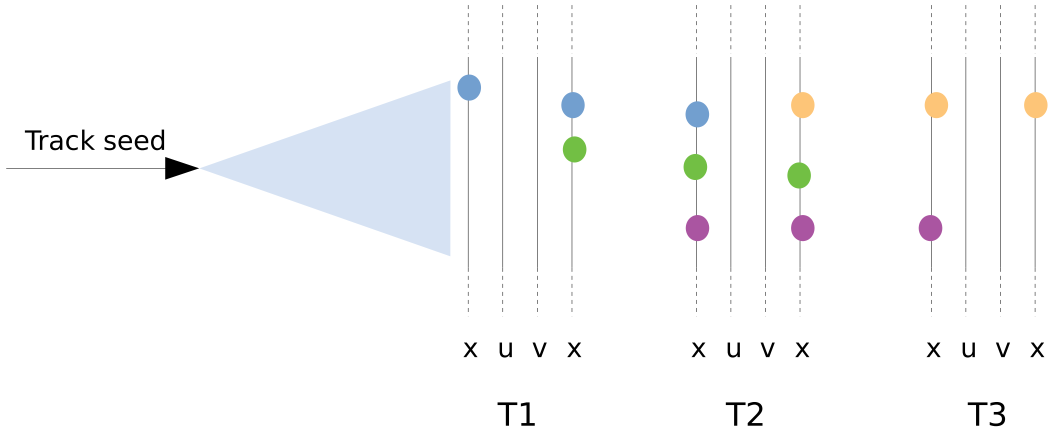


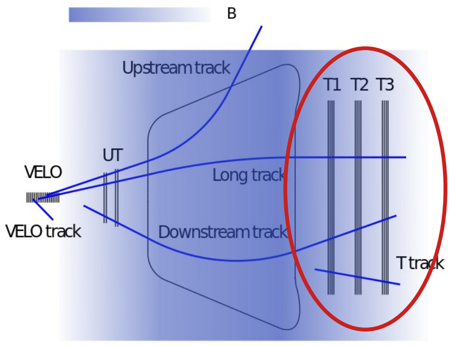
- Primary vertices (Pvs) extended along beamline
- Histogram of track z-positions at beamline
- Clusters in histogram → PV candidates
- Fill histogram in parallel
- Every track contributes to every PV candidate with a weight → no inter-dependence among PV candidates
- PV candidate fitting parallelized across
  - PV candidates
  - Tracks

SciFi track reconstruction

- 12 layers of scintillating fibres
- xuvx configuration
- Build seeds of triplets in different combinations of layers in parallel
  → avoid inefficiencies due to fibre inefficiency
- Extend seeds in parallel
- Use parameterization of trajectories inside magnetic field rather than lookup in field map
- Reconstruct momentum based on bending between Velo and SciFi part of the track

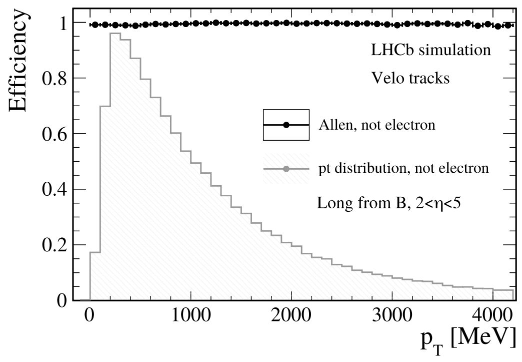


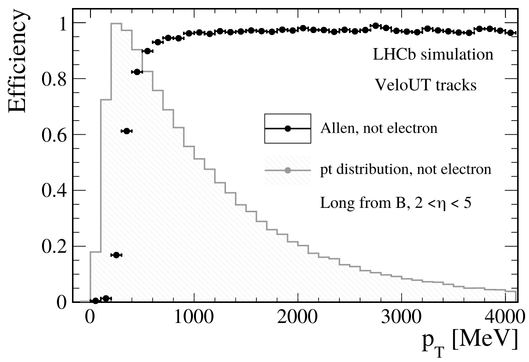


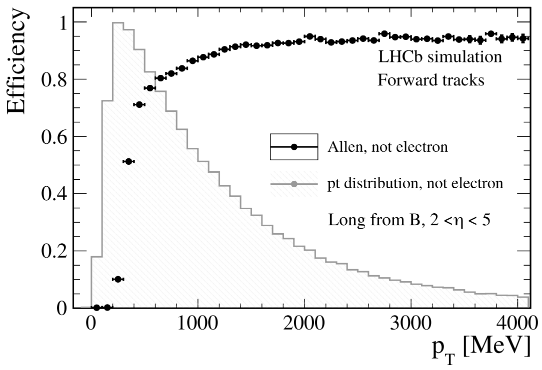


Physics performance: Track reconstruction

Track reconstruction efficiency

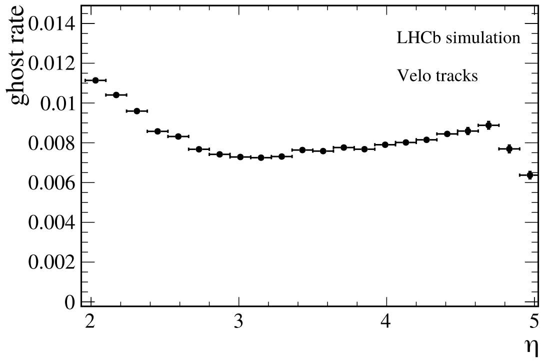


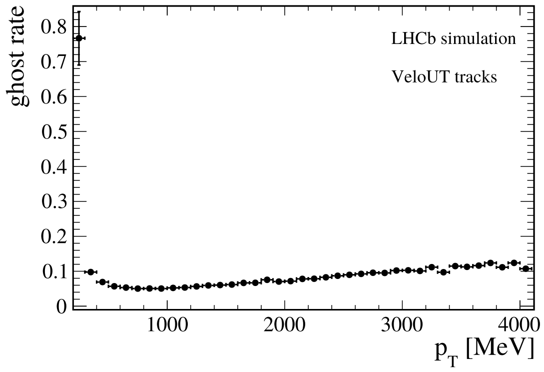


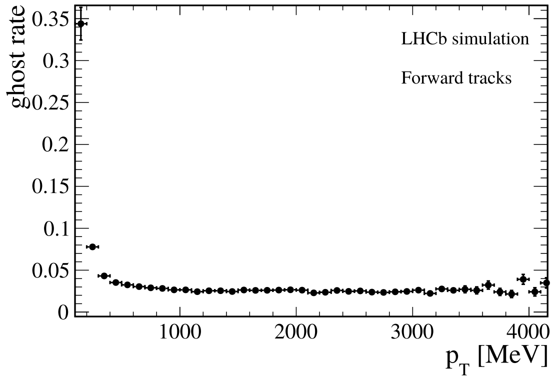


Fake rate

LHCb-FIGURE-2020-014



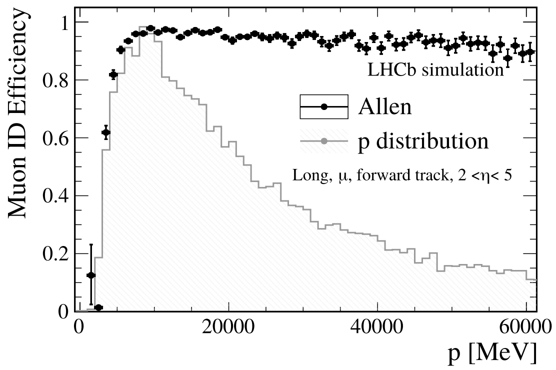




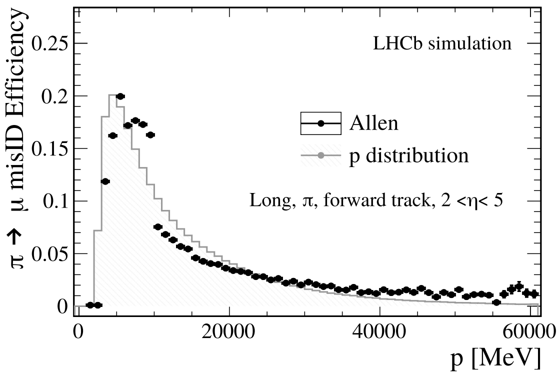
Track reconstruction @ 30 MHz on GPUs very successful

Physics performance: Muon ID, PVs, resolution

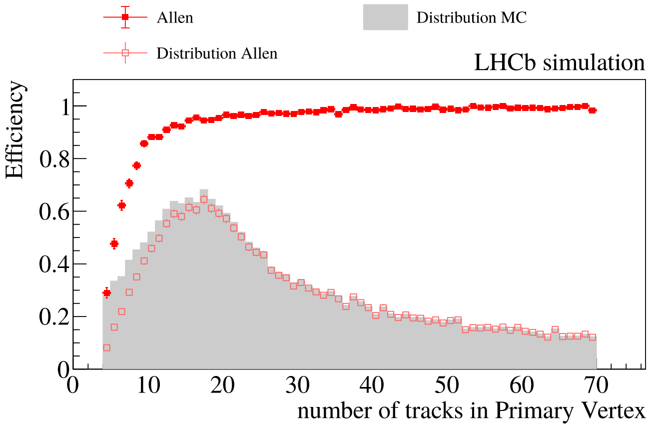
Muon ID efficiency



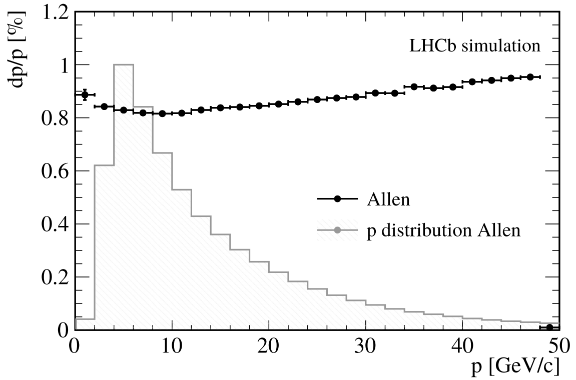
π→μ mis-ID efficiency



PV reconstruction efficiency



Momentum resolution



LHCb-FIGURE-2020-014

HLT1: Trigger rates

Monitoring & calibration lines

Trigger	Rate [kHz]
ErrorEvent	0 ± 0
PassThrough	30000 ± 0
NoBeams	5 ± 3
BeamOne	18 ± 5
BeamTwo	8 ± 3
BothBeams	4 ± 2
ODINNoBias	0 ± 0
ODINLumi	1 ± 1
GECPassthrough	27822 ± 52
VeloMicroBias	26 ± 6
TrackMVA	409 ± 23
TrackMuonMVA	23 ± 6
SingleHighPtMuon	7 ± 3
TwoTrackMVA	503 ± 26
DiMuonHighMass	131 ± 13
DiMuonLowMass	177 ± 15
DiMuonSoft	8 ± 3
D2KPi	93 ± 11
D2PiPi	34 ± 7
D2KK	76 ± 10
Total w/o pass through lines	1157 ± 39

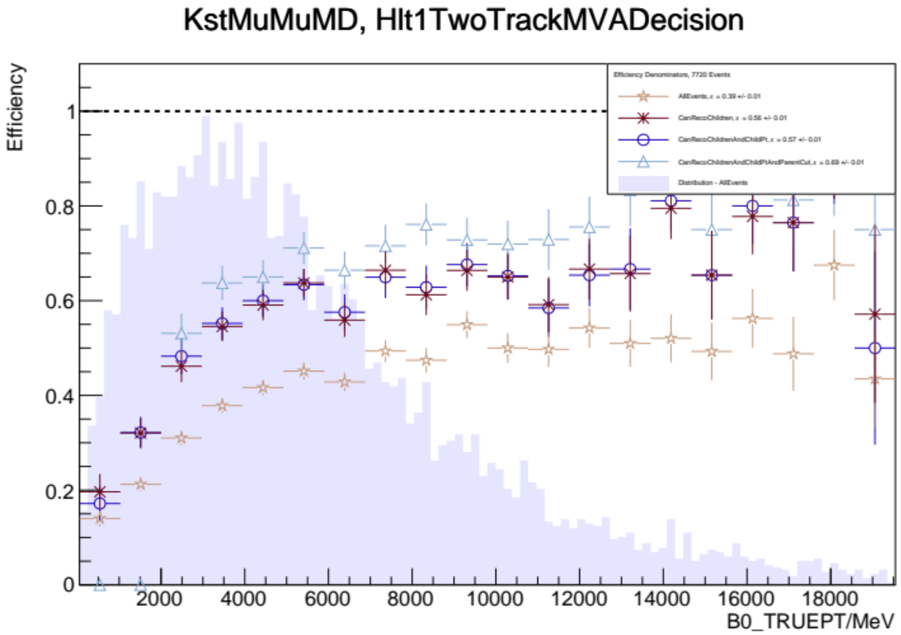
Physics selections

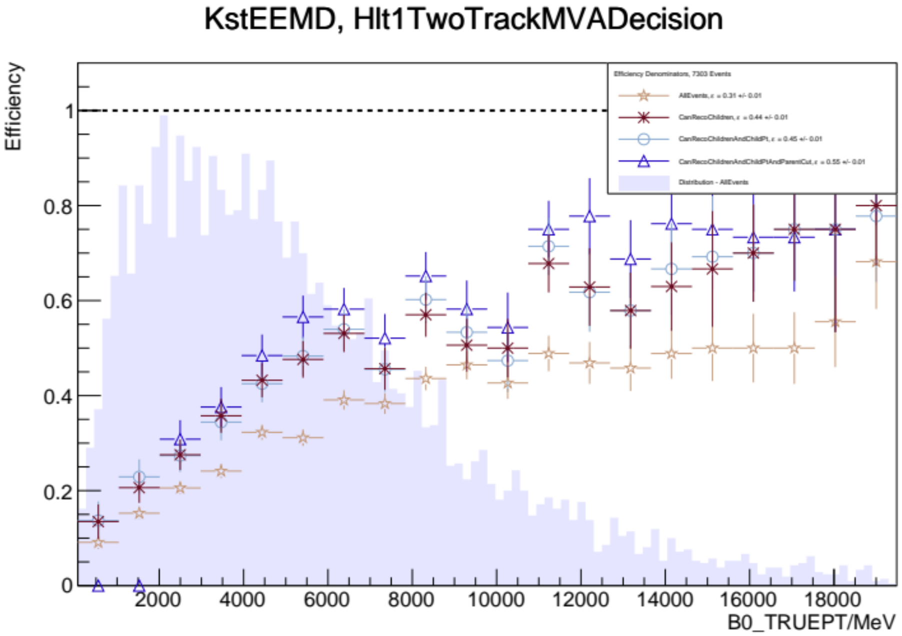
Alignment

Event rate reduced by factor 30

LHCb-FIGURE-2020-014

HLT1: Selection efficiencies





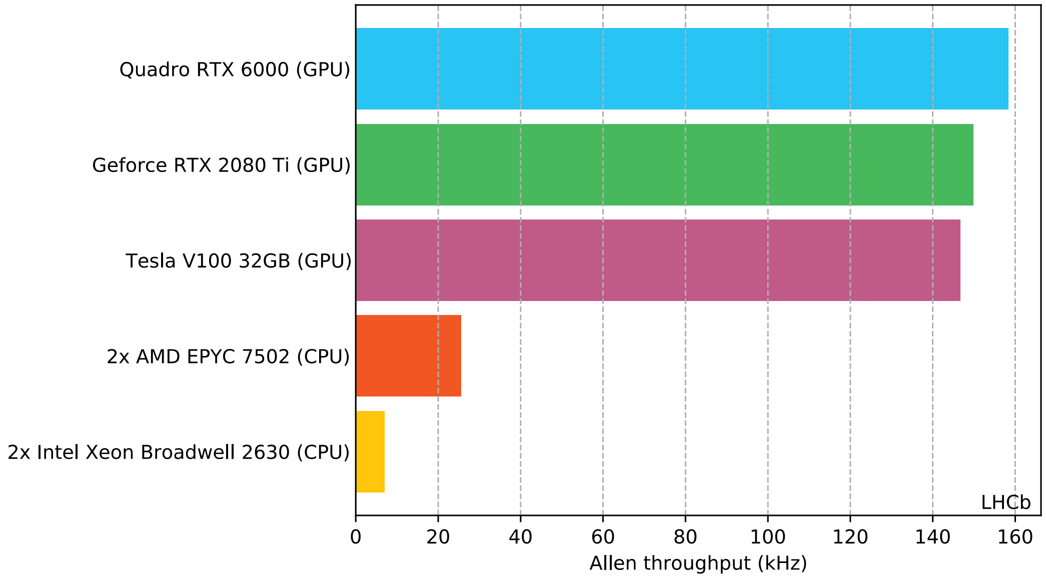
CERN-LHCC-2020-006

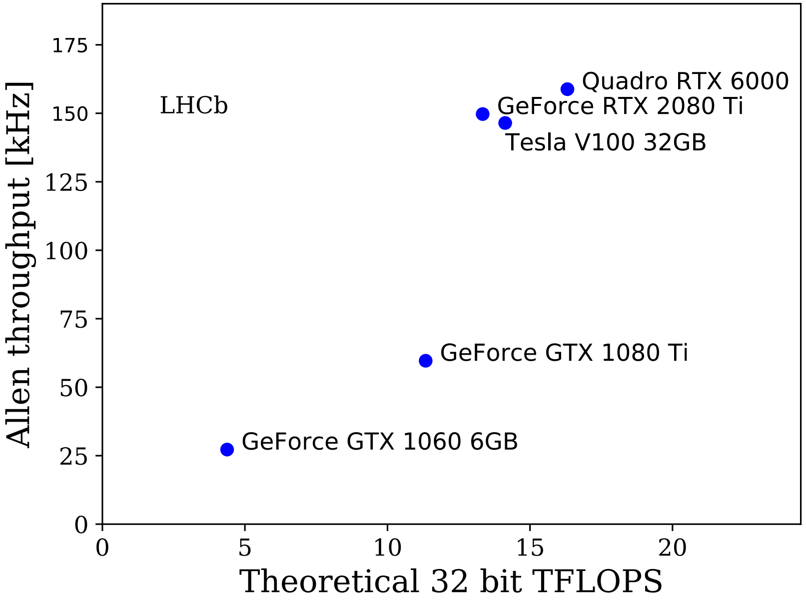
Selection efficiencies for electron and muon final states similar

In Run 2: Electron selection efficiency roughly factor two worse than muons due to hardware level trigger

Computing performance

LHCb-FIGURE-2020-014

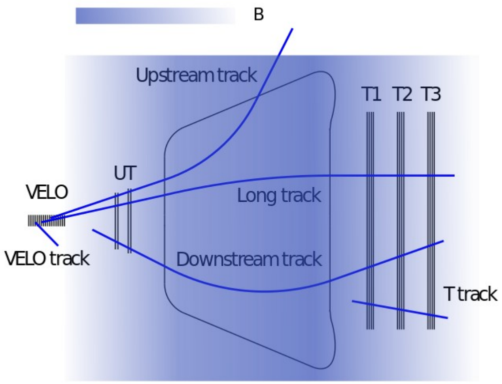


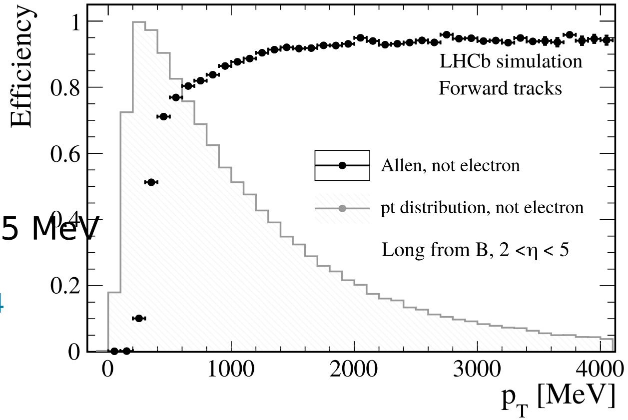


- Require about 215 GPU cards to process full HLT1 @ 30 MHz
- Have slots for 500 cards
- Computational performance scales well with GPU generations → expect improvements with next generation cards (coming out this year)

Possible add-ons to the HLT1

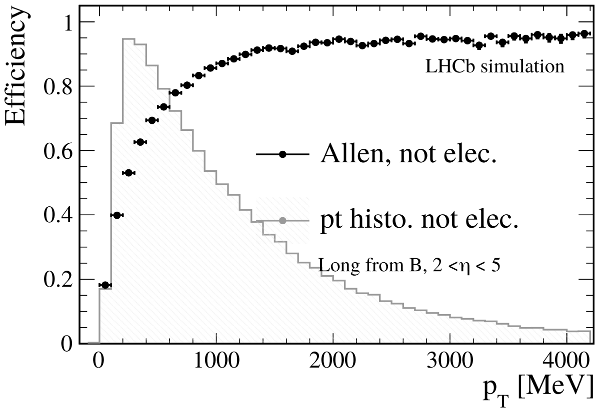
- Large headroom in throughput of “standard” GPU HLT1
  → Can think of more efficient settings & additional algorithms, such as:
- Track reconstruction w/o cut on p_T (especially beneficial for D decays)
- No global event cut (removing the 10% busiest events) for some algorithms (for example to reconstruct high pT muons for electroweak physics)
- Calorimeter reconstruction → electron ID
- Downstream track reconstruction for long-lived particles





p > 2.5 GeV, p_T > 425 MeV

LHCb-FIGURE-2020-014



p > 3 GeV, p_T > 0 MeV

CERN-LHCC-2020-006

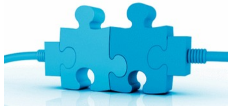
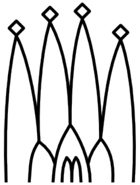
The Allen project

- Fully standalone software project: https://gitlab.cern.ch/lhcb/Allen
- Framework developed for processing HLT1 on GPUs
- Runs on CPU, Nvidia GPUs (CUDA, CUDACLANG), AMD GPUs (HIP)
- GPU code written in CUDA
- Cross-architecture compatibility via macros (HIP, CPU)

- Named after Frances E. Allen

Allen software framework

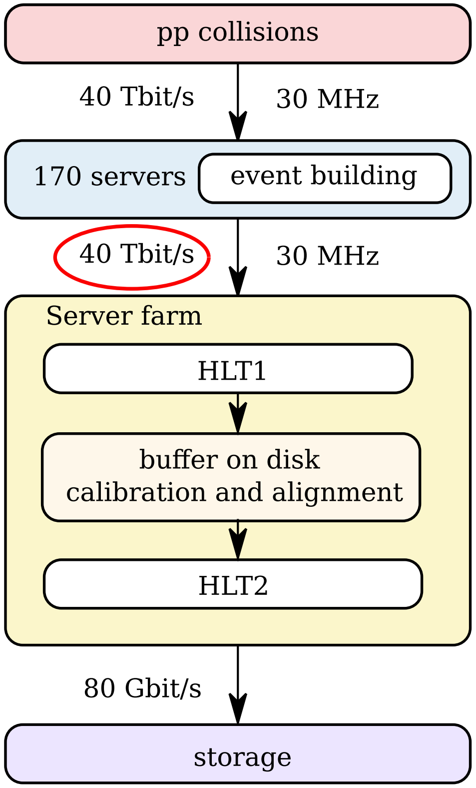
- Algorithm sequences defined in python and generated at compile time
    - Algorithms to run with inputs / outputs, properties (minimum momentum cut-off etc.)
- Memory manager:
    - Large chunk of GPU memory allocated at start-up
    - Pieces of memory assigned to algorithms by memory manager
    - Memory size has to be known at compile time
- Cross-architecture compatibility via macros & few coding guide lines
- Support three modes:
    - Standalone project
    - Compiling with Gaudi for data acquisition
    - Compiling with Gaudi for simulation workflow and offline studies
- Allen-Gaudi workshop took place in July
    - Viewpoints on heterogeneity from all four LHC experiments & WLCG
    - How scheduling and memory management of Allen and Gaudi can function together

History: HLT1 architecture choice

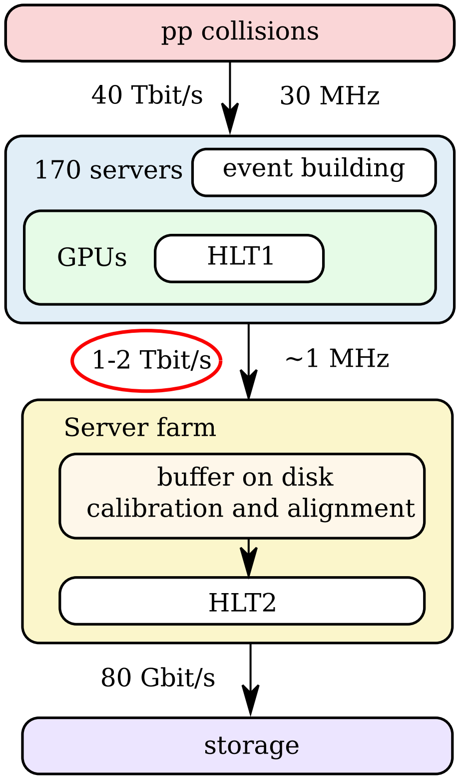
Proposal in TDR (2014)

CERN-LHCC-2014-016



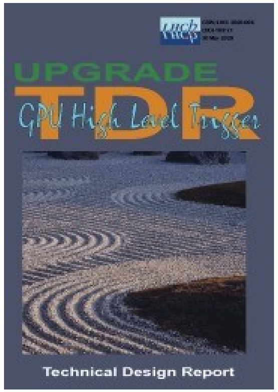
Updated strategy (as of 5/2020)

CERN-LHCC-2020-006

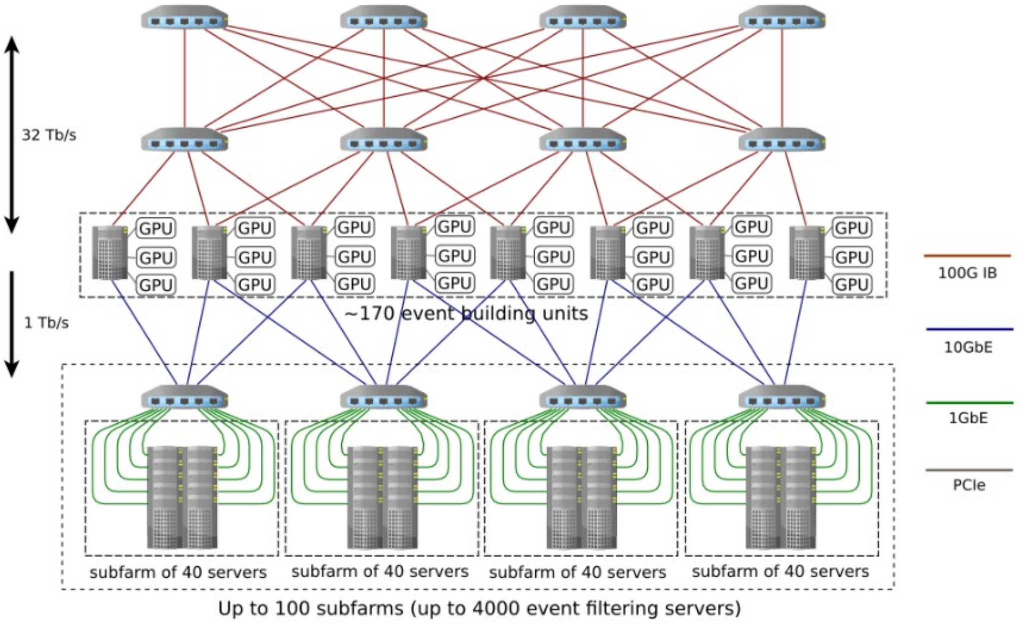


- Developed two solutions simultaneously
- Both the multi-threaded CPU & the GPU HLT1 fulfilled the requirements from the 2014 TDR
- LHCb was in the luxury situation to choose among them
- Compared physics performance & price-performance

→ decided for GPU solution



Future: Towards commissioning



- Communication with event builder network
  - Data packet format of input
  - Passing output of HLT1 to HLT2
- Final data formats of sub-detectors
- Monitoring: histograms, counters
- Communication with geometry description (DD4Hep)
- As sub-detectors are commissioned, run algorithms on first data
  - Cosmic tracks
  - Calorimeter clusters (sources)

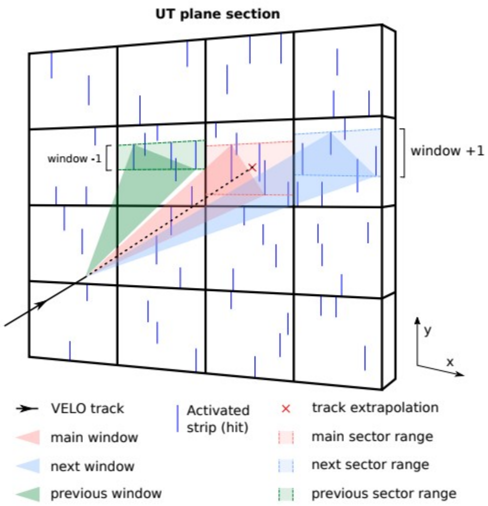
Summary

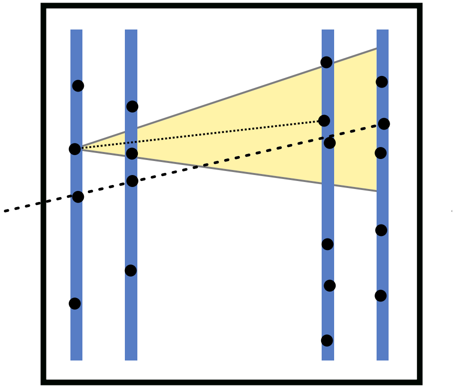
- LHCb is undergoing a major upgrade to push the intensity frontier
- Efficient real-time data selection is key to exploiting the full physics potential
- LHCb is commissioning the first complete high-throughput GPU trigger for an HEP experiment
- Many options to improve LHCb's physics potential by adding to the "basic" HLT1 reconstruction sequence thanks to large headroom in computing performance
- Economically sustainable trigger (save money due to reduced network between event builders and filter farm)
- With a heterogeneous trigger LHCb can benefit from future industry developments
- GPU developments result in valuable training for young scientists

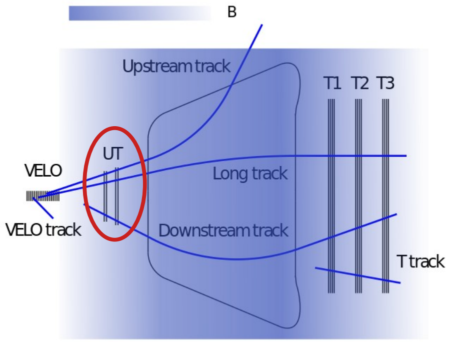
Backup

UT track reconstruction

- Four layers of silicon strip detectors
- Extrapolate Velo tracks to the UT planes based on lookup-table for minimum momentum requirement → parallelize across tracks
- Decode UT hits into memory layout optimized for fast lookup around extrapolated track position
- Look for stubs in the UT detector → parallelize across combinations of two hits
- Match Velo seeds to stubs in the UT → parallelize across Velo tracks





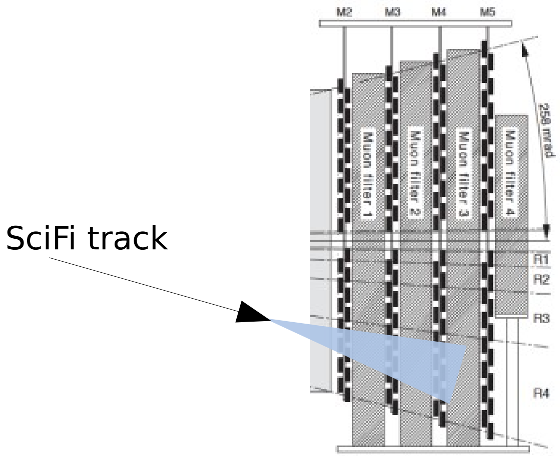


P. Fernandez Declara, D. Campora Perez, J. Garcia-Blas, D. vom Bruch, J. Daniel Garca, N. Neufeld , IEEE Access 7 (2019)

Muon identification & track fit

Muon identification

- Extrapolate SciFi tracks into muon chambers
- Match track to hits
- Parallelize across tracks and muon chambers



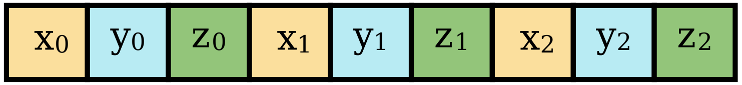
- **Track fit: Kalman filter**
- Goal: Improve track description close to the beamline for precise determination of impact parameter
- Only fit part of the track within the Velo detector
- Use parameterized Kalman filter → no need for magnetic field map and detector material description
- Showed that it works well in single precision

How to make best use of the TFLOPs on a GPU

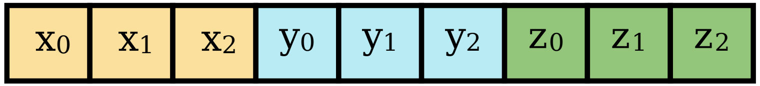
- Design algorithm for extreme parallelism (thousands of threads active)
- Assign paths with branches to different thread blocks
- Keep similar paths in the same thread block
- Prefer linear over iterative algorithms
- Port chains of algorithms
- Avoid data copies
- Or hide memory transfers
- Make GPU workflow asynchronous with respect to the CPU
- Explore and use the minimal floating point precision required by the algorithm
- Don't be afraid to redesign data structures
- Reuse preallocated memory (no dynamic memory allocations on the GPU)
- Minimize memory footprint

Data structures

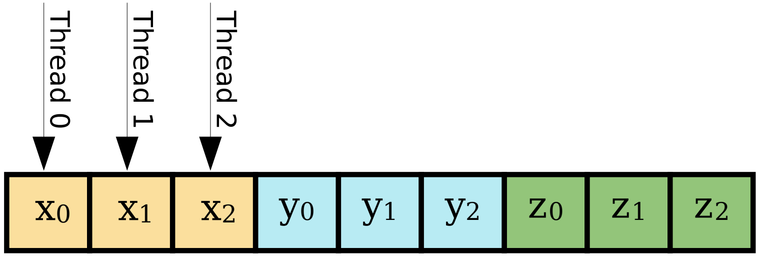
Array of structures



Structure of arrays

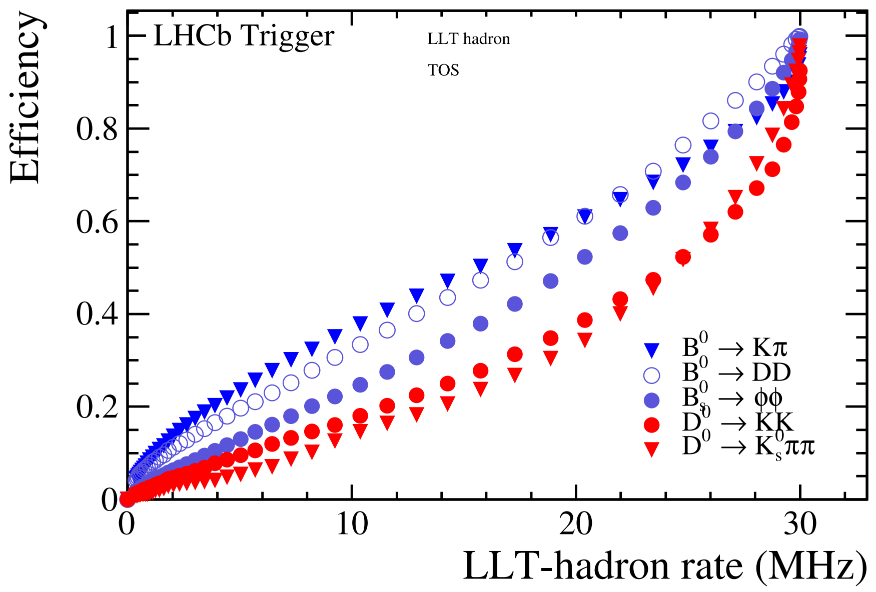


- GPU memory bandwidth best exploited with coalesced memory access
- Use Structure of Arrays (SoA) data layout
- Decoded raw data can directly be stored in SoA format
- Reconstructed tracks, vertices etc. are also stored in SoAs
- Only requirement: need to know the array size at memory allocation time

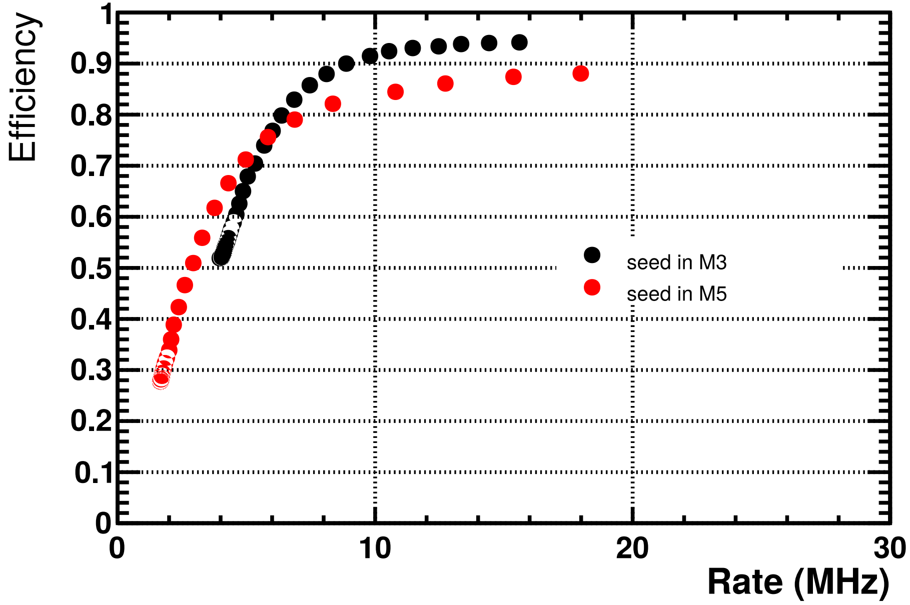


Why no low level trigger?

Low level trigger on E_T from the calorimeter

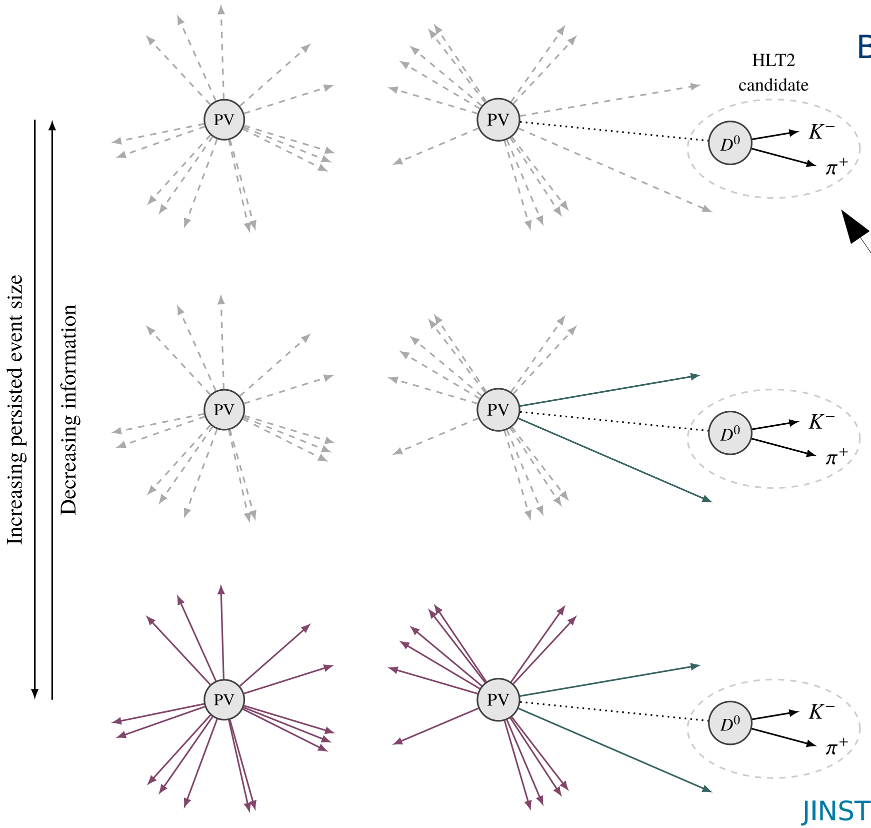


Low level trigger on muon p_T, $B \to K^{*}\mu\mu$



Need track reconstruction at first trigger stage

Selective persistency



Bandwidth [MB/s] ~ Trigger output rate [kHz] x average event size [kB]

- Trigger *bandwidth* is crucial, not trigger rate
- Real-time selection occurs with offline quality
- Only store high-level objects reconstructed in real-time
- Reduced event format → reduction of event size
  → higher efficiency for same bandwidth
- “Turbo stream”
- High degree of flexibility:
  - Only objects used in trigger selection
  - Objects used in trigger selection & user-defined selection
  - All reconstructed objects
- Raw data only stored in calibration stream

JINST 14 (2019) P04006

Framework requirements

Support various architectures

Low entry point for user

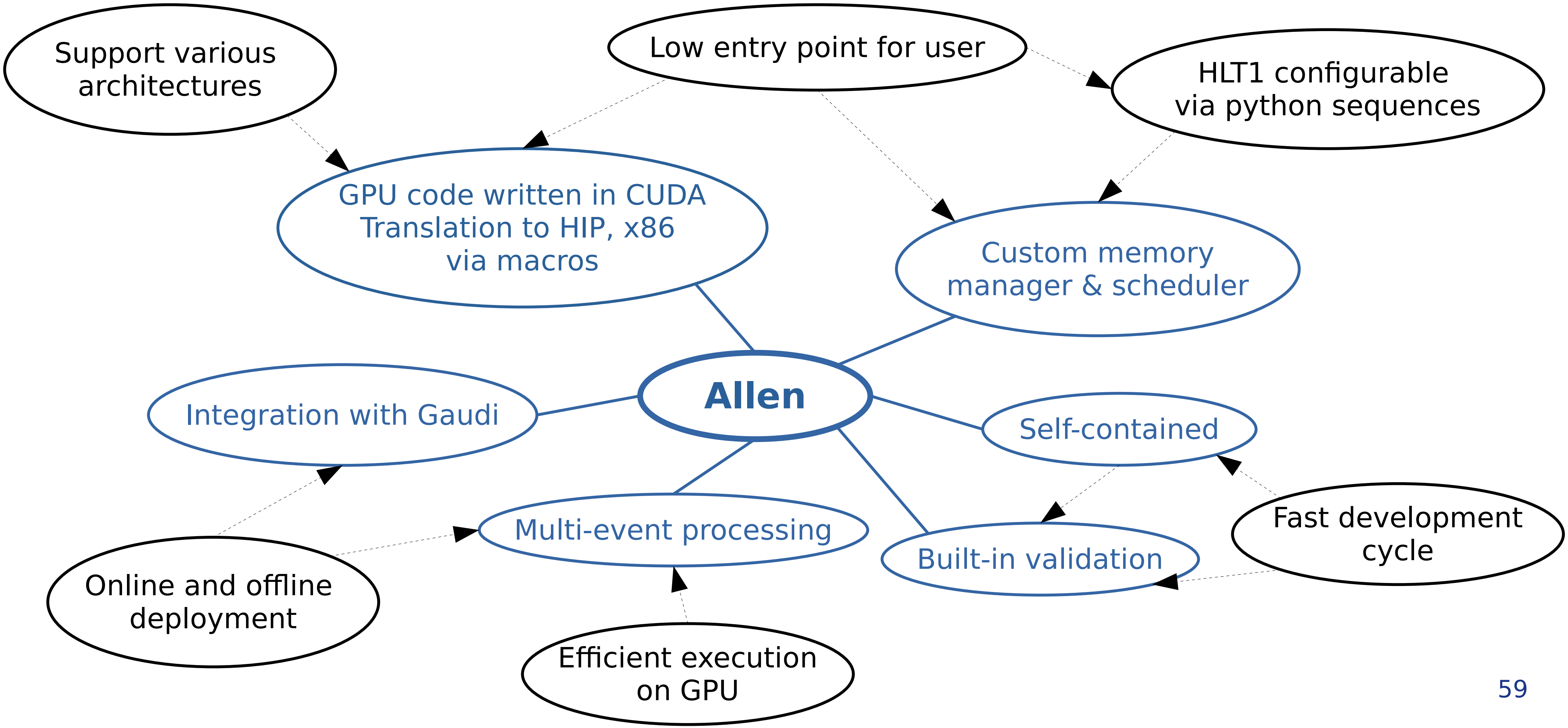
HLT1 configurable via python sequences

Fast development cycle

Online and offline deployment

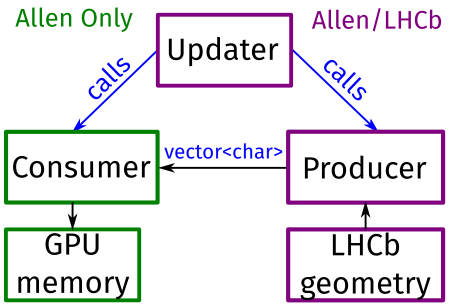
Efficient execution on GPU

Framework design



Online integration

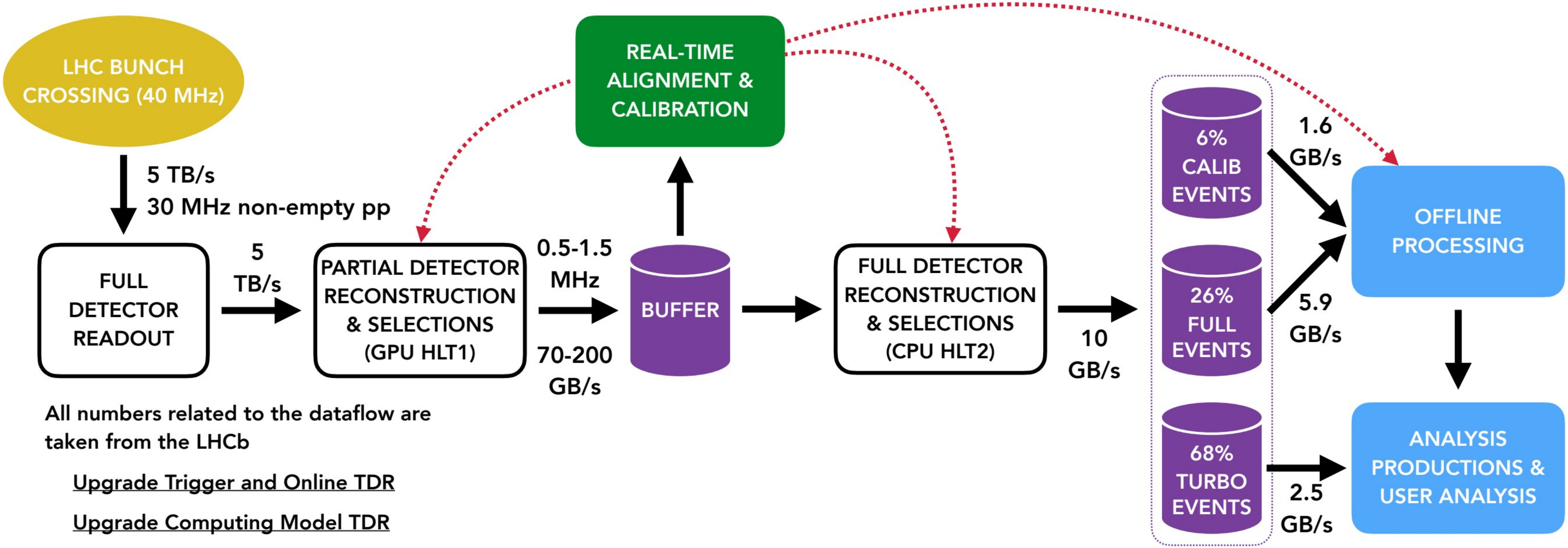
- Event-loop steered by Allen in multi-event batches
- Non-event data requested from Gaudi upon run change
    - Aligned & calibrated detector description
    - Magnet polarity
    - Special running conditions
- Raw data from selected events + decision reports sent to HLT2



Offline integration

- For simulation & offline studies
- Use x86 compilation of Allen → can run on the WLCG
- Event loop steered by Gaudi
- Allen called one event at a time

Data flow in Run 3



All numbers related to the dataflow are taken from the LHCb

- Upgrade Trigger and Online TDR
- Upgrade Computing Model TDR