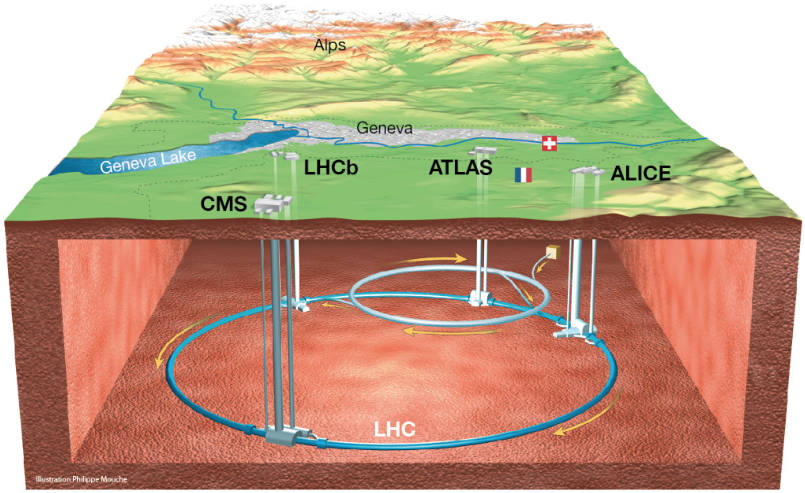
High-Throughput Real-time Data Processing with GPUs at the LHCb experiment at CERN

Daniel Hugo Cámpora Pérez, Roel Aaij
dcampora@cern.ch, roelaaij@nikhef.nl

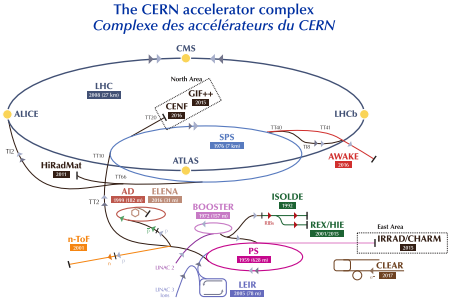
GAHTIe Kickoff, September 9th, 2021

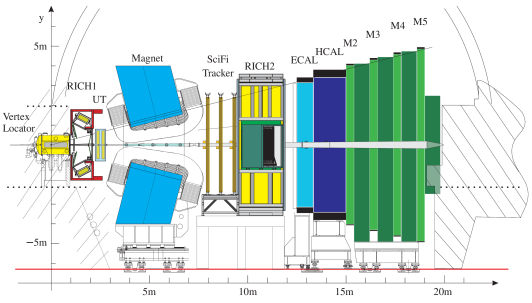
Maastricht University
Nikhef
CERN

Introduction

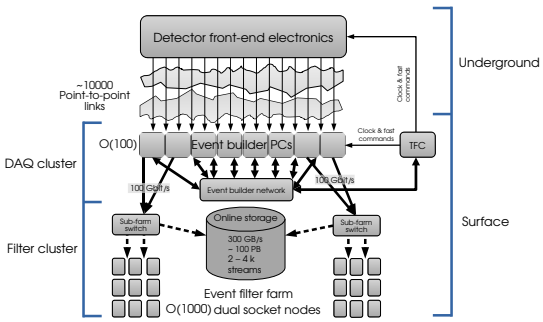

Alps
Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
LHC
Illustration Philippe Mouche

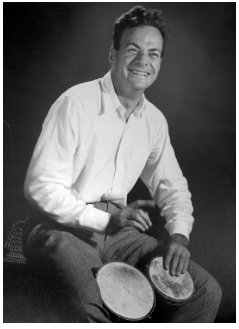
A particle's journey

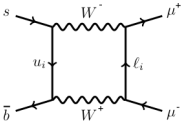




At the end of the day

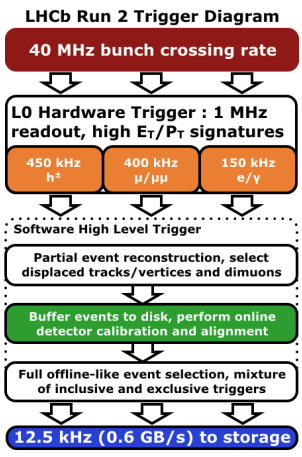




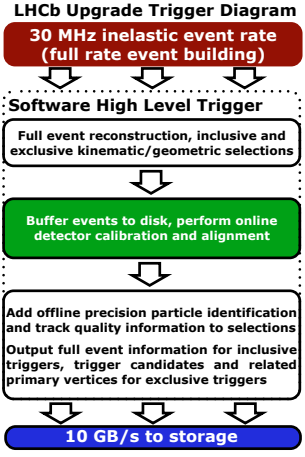


Processing LHC Data (https://www.youtube.com/watch?v=jDC3-QSiLB4)

A change of paradigm

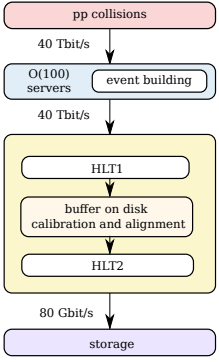


2017



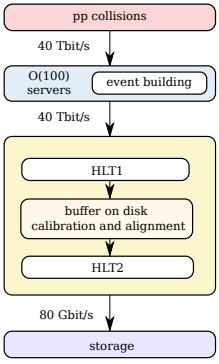
2021

Trigger and Data Acquisition system with GPUs

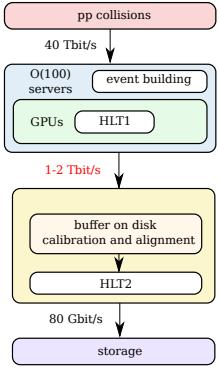


Baseline DAQ

Trigger and Data Acquisition system with GPUs



Baseline DAQ



GPU-equipped DAQ

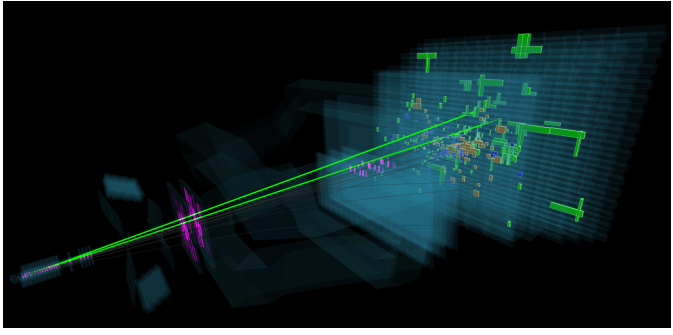
By placing GPUs in the Event Builders, the cost of the network to the Event Filter Farm is reduced.

What do we do with the data?

Event reconstruction

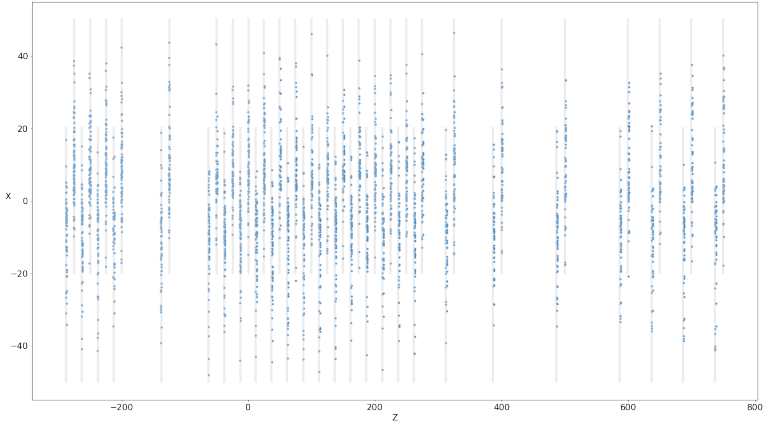
We cannot keep all the data however, we must filter it.

For that reason, we **reconstruct** what happened in every collision *event*.



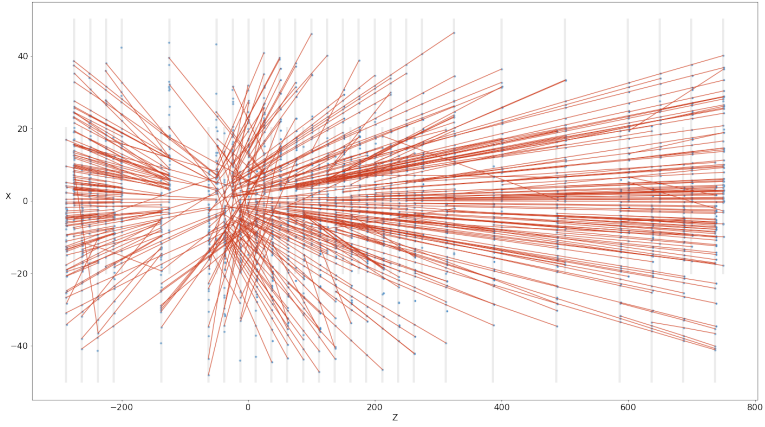
Pattern recognition - Tracking

Tracking – Reconstructing the trajectory of a particle from *hits* in its path. Yields information about p and trajectory.



Pattern recognition - Tracking

Tracking – Reconstructing the trajectory of a particle from *hits* in its path. Yields information about p and trajectory.



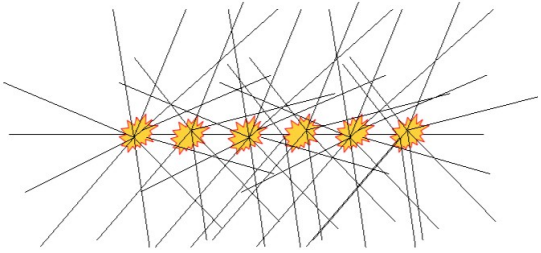
Pattern recognition - Tracking

Tracking – Reconstructing the trajectory of a particle from *hits* in its path. Yields information about p and trajectory.

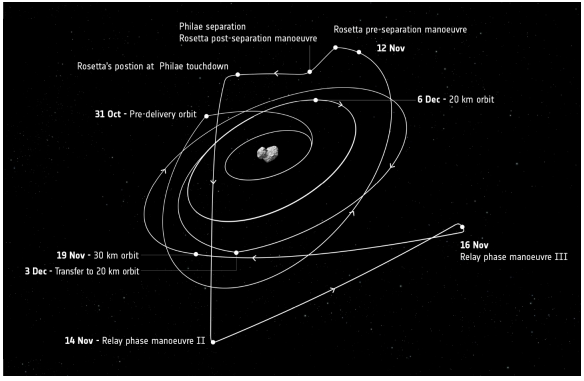
Tracks can bend in the presence of a magnetic field.

Pattern recognition - Vertices

Vertex finding – Reconstruct vertices of collision and particle decays.



Kalman filter – Estimate particle trajectories according to a mathematical model describing it and empirical data.



Rosetta path (©ESA).

[Massive] parallelism

In LHCb alone, 40 Tbits per second will be processed by 2021.

These data are structured as follows:

- 30 million independent collisions **events** per second, 100 kB each.

For each event:

- Four tracking problems – **O(100)** tracks per event each.
- Vertex finders – **O(100)** vertices per event.
- Kalman filter – **O(100)** instances per event.

[Massive] parallelism

In LHCb alone, 40 Tbits per second will be processed by 2021.

These data are structured as follows:

- 30 million independent collisions **events** per second, 100 kB each.

For each event:

- Four tracking problems – **O(100)** tracks per event each.
- Vertex finders – **O(100)** vertices per event.
- Kalman filter – **O(100)** instances per event.

Looks like a problem where massively parallel architectures can make a difference!

Real-time reconstruction on GPUs

Demonstrators

In the field of GPU computing, it is common to develop demonstrators to prove the technology is viable to solve a problem.

In High Energy Physics, the two figures of merit are:

- **Physics efficiency**
- **Throughput**

Demonstrators

In the field of GPU computing, it is common to develop demonstrators to prove the technology is viable to solve a problem.

In High Energy Physics, the two figures of merit are:

- **Physics efficiency**
- **Throughput**

Turns out a key ingredient to achieving *good throughput* is to support multiple events[1].

[1] D. H. Cámpora Pérez. "Optimization of high-throughput real-time processes in physics reconstruction". PhD thesis. University of Sevilla, Spain.

Allen

The Allen framework [2] is a modular, scalable and flexible framework that supports multiple-event execution and is geared towards GPUs.

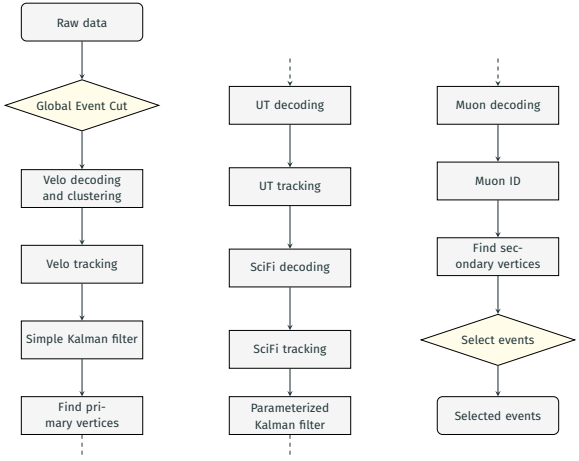
- Built on top of C++17

Features:

- Supports CUDA, CPU and HIP targets.
- Multi-threaded framework.
- Configurable static and pipelined sequences.
- Custom memory manager, no dynamic allocations, SOA datatypes.
- Built-in validation. Generation of graphs with ROOT.

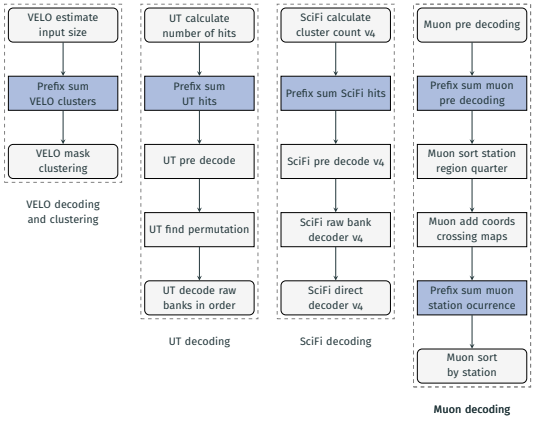
[2] R. Aaij *et al.* "Allen: A high level trigger on GPUs for LHCb". Computing and Software for Big Science. To appear.

Sequence of algorithms implemented in Allen

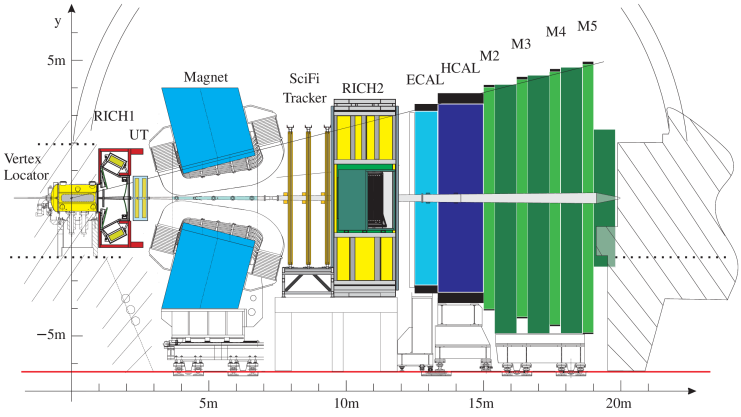


Data decoding

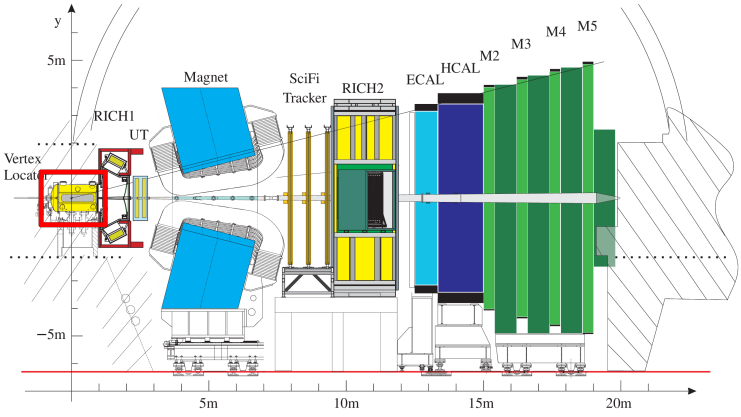
Raw data must be **decoded** first. Prefix sums are used to determine offsets / sizes of data buffers in each event. We *offload* them to CPU (blue boxes).



The LHCb first stage reconstruction – step by step

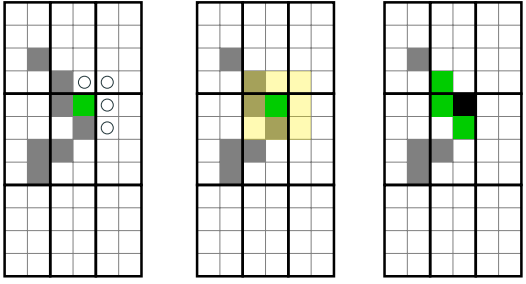


The LHCb first stage reconstruction – VELO reconstruction

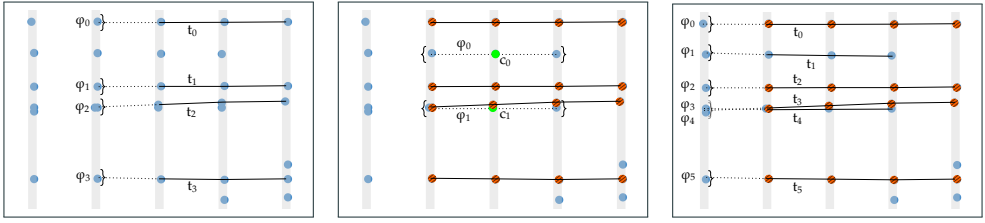


Clustering

VELO detector modules are traversed seeking **8-connected groups of pixels**. Average x, y coordinates are saved (a variant of CCA). We developed a *mask clustering* method.



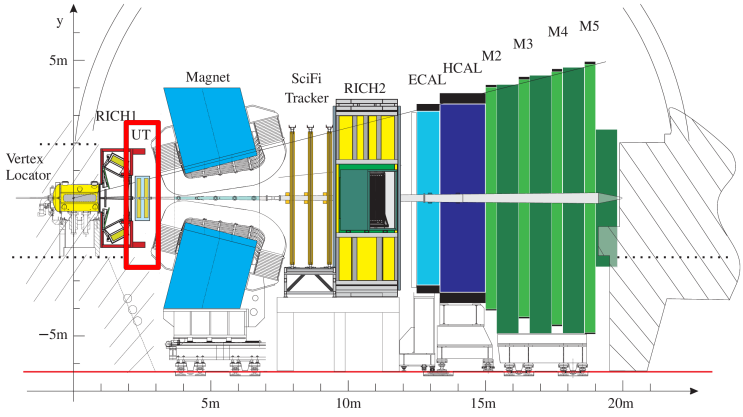
Hits are traversed using a **parallel local track forwarding** method [3]. Track seeds are created, forwarded, and used hits are flagged, in a predictable iterative pattern.



Seeding and Forwarding #0 | Seeding #1 | Forwarding #1

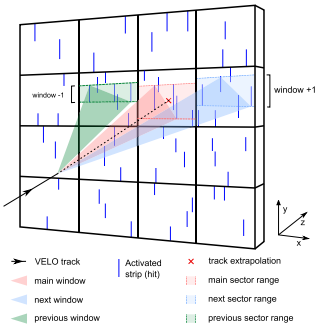
[3] D. H. Cámpora Pérez, N. Neufeld, and A. Riscos Núñez. "A Fast Local Algorithm for Track Reconstruction on Parallel Architectures". In: *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2019, pp. 698–707.

The LHCb first stage reconstruction – UT reconstruction



UT tracking

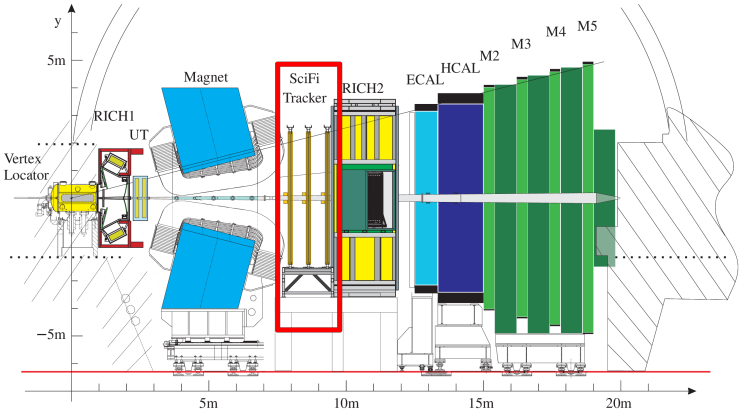
Tracks formed at the VELO subdetector are **extended** using UT hits. A 2D-structure is used in conjunction with search windows to find the best hits in an efficient manner [4].



Search window.

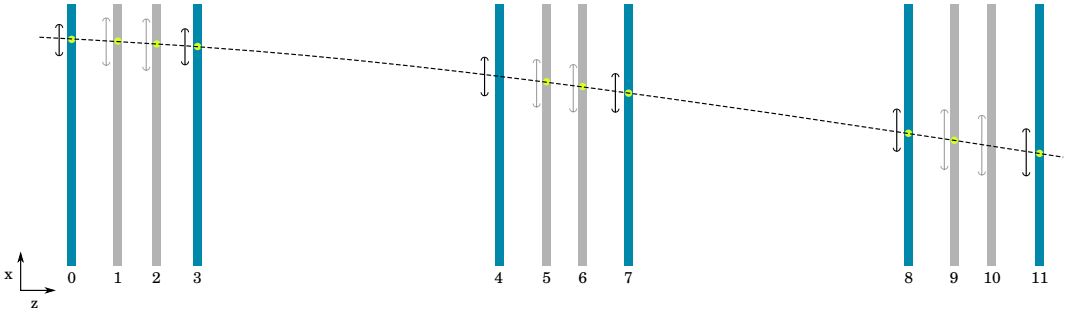
[4] P. Fernandez Declara et al. "A parallel-computing algorithm for high-energy physics particle tracking and decoding using GPU architectures". In: *IEEE Access* (2019), pp. 91612–91626.

The LHCb first stage reconstruction – SciFi reconstruction



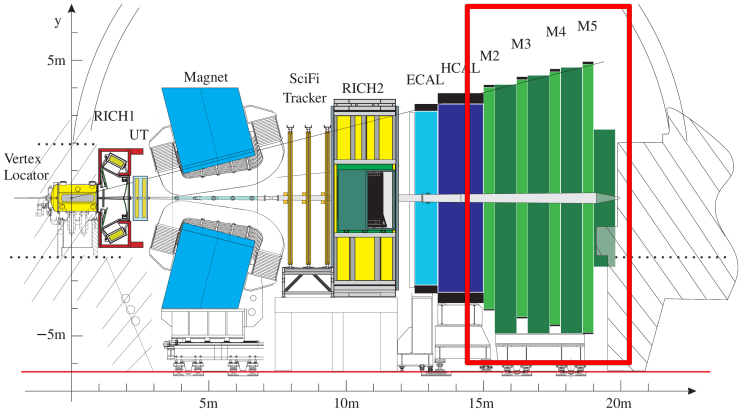
Forward tracking

Tracks are **extended once again** using SciFi hits, creating *forward tracks*. A model of the magnetic field is used to estimate the location of tracks, and search windows are traversed seeking compatible triplets.



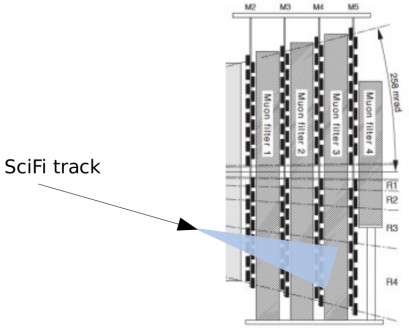
Compatible triplets are sought in either $\{0, 4, 8\}$ or $\{3, 7, 11\}$.

The LHCb first stage reconstruction – Muon stations



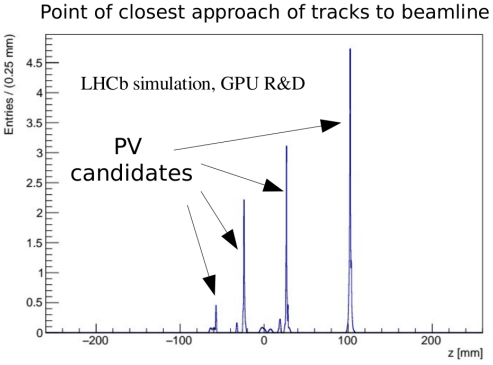
Muon tracking

Tracks are **extended one last time** using hits from the muon stations. Tracks found in this manner are identified as *muons*.



Vertex finding

Primary vertices (from *particle collisions*) are found using VELO tracks, and **secondary vertices** (from *particle decays*) are found using forward tracks. A histogramming method is employed.



Kalman filter

A Kalman filter is run over the tracks to **improve their resolution**.

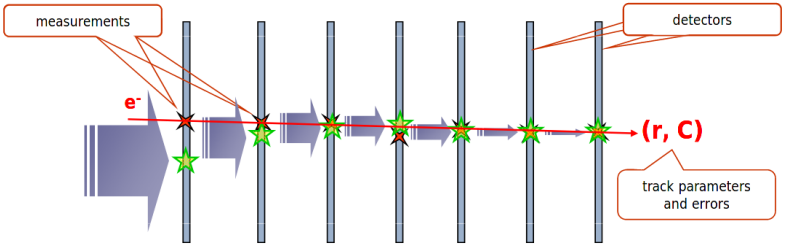


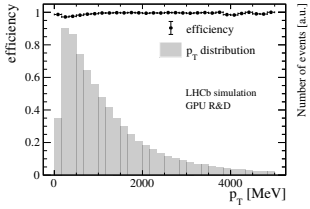
Image from *Event Reconstruction in High Energy Physics Experiments*, I. Kisel.

Selections

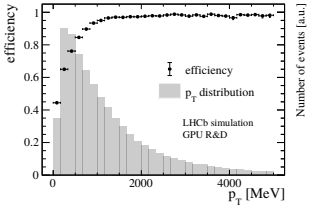
Finally, a **set of filters** is applied over the resulting data. Only those events passing at least one filter are kept, and the rest are discarded.

Validation

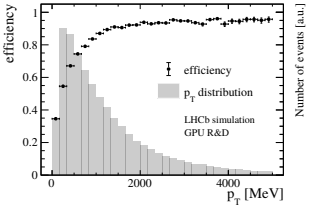
The complete sequence has been **validated** using Monte Carlo simulated data. The resulting performance *fulfills the requirements* imposed by the physics programme of LHCb.



VELO



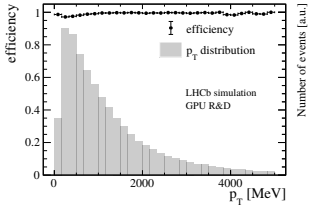
UT



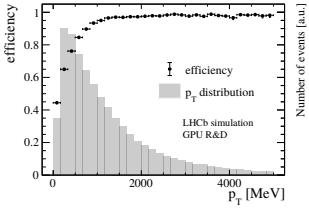
Forward

Validation

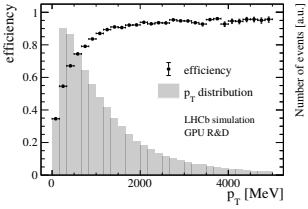
The complete sequence has been **validated** using Monte Carlo simulated data. The resulting performance *fulfills the requirements* imposed by the physics programme of LHCb.



VELO



UT



Forward

So what about throughput?

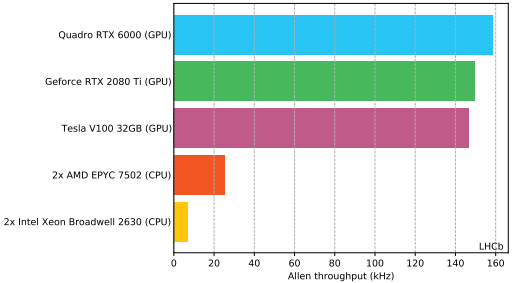
Performance and integration

Methodology

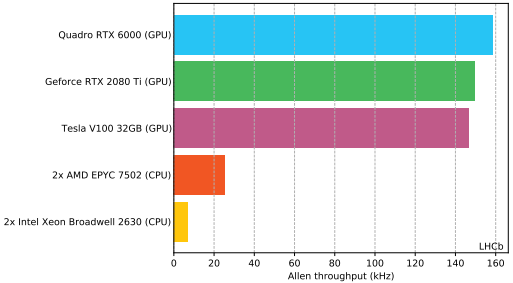
We are looking at various GPUs. Here is a table with their main features:

Feature	Geforce RTX 2080 Ti	Quadro RTX 6000	Tesla T4	Tesla V100
# cores	4352 (CUDA)	4608 (CUDA)	2560 (CUDA)	5120 (CUDA)
Max freq.	1.545 GHz	1.77 GHz	1.59 GHz	1.37 GHz
Cache (L2)	6 MiB	6 MiB	6 MiB	6 MiB
DRAM	10.92 GiB GDDR6	24 GiB GDDR6	16 GiB GDDR6	32 GiB HBM2
CUDA capability	7.5	7.5	7.5	7.0
TDP	250W	250W	70W	250W

Throughput


Quadro RTX 6000 (GPU)
Geforce RTX 2080 Ti (GPU)
Tesla V100 32GB (GPU)
2x AMD EPYC 7502 (CPU)
2x Intel Xeon Broadwell 2630 (CPU)
LHCb
0
20
40
60
80
100
120
140
160
Allen throughput (kHz)

Throughput



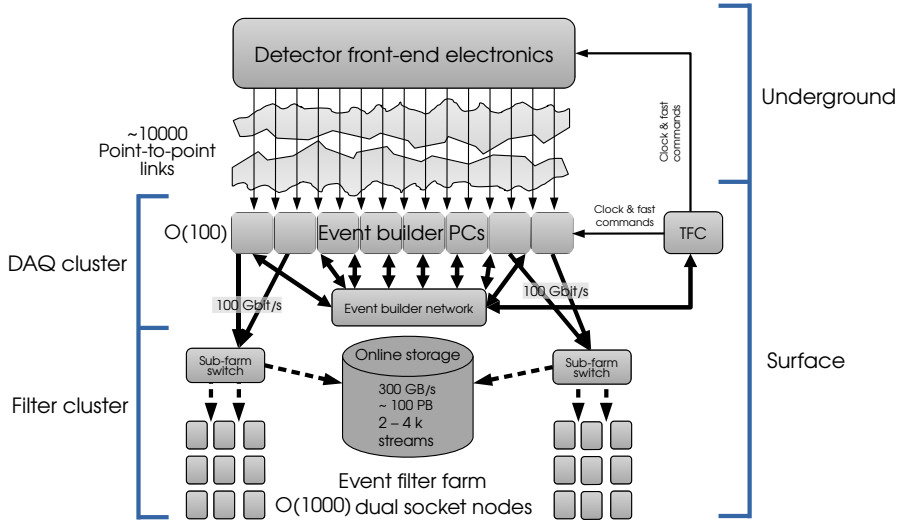
The target throughput is 30 MHz, and there is room for up to 500 GPU cards. The target is amply fulfilled with the top three cards.

Integration

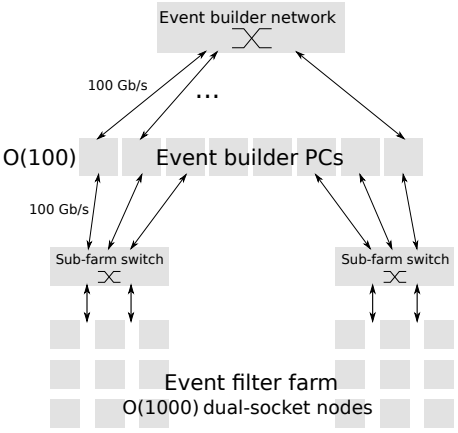
In order to integrate GPUs in LHCb's Data Acquisition system, several **technical questions** remain, such as:

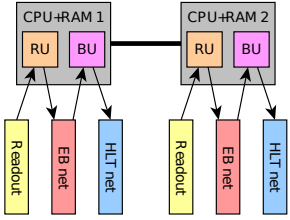
- Where would we place the GPUs physically?
- Would the system be compromised if GPUs are placed?
- Is it cost-effective to deploy GPUs?

Surely you remember the Data Acquisition system



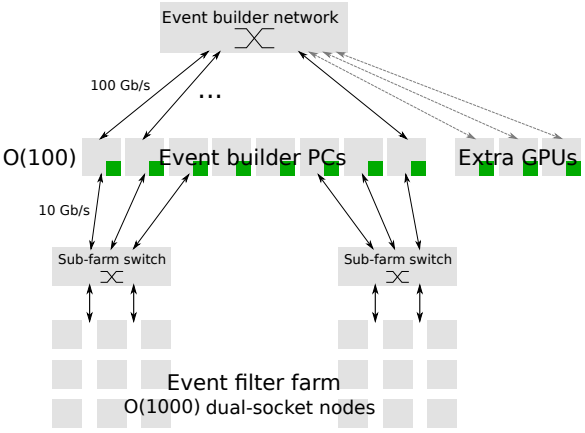
Baseline Data Acquisition system

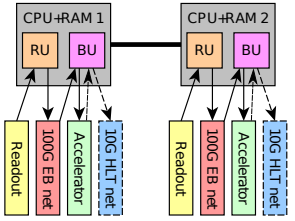




Event builder PC.

Data Acquisition system with GPUs in Event Builders



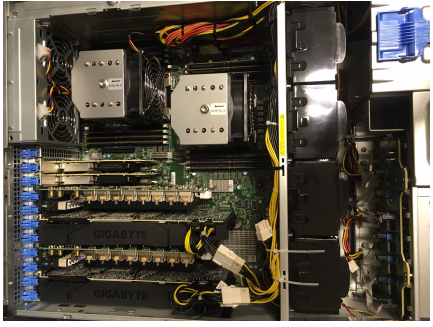


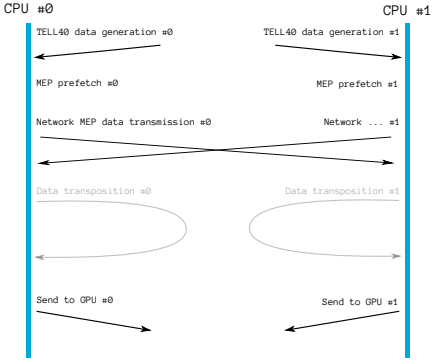
GPU-equipped event builder PC.

Integration tests

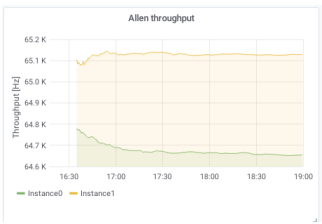
The most attractive realization of a GPU HLT1 in LHCb would be in the Event Builders.

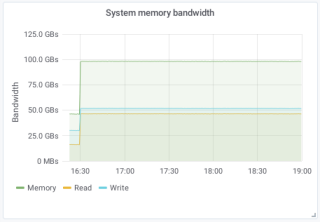
Many aspects need to be demonstrated, such as CPU consumption, memory consumption and throughput, airflow, thermal stability, GPU performance stability, network throughput...

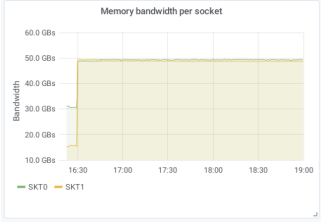


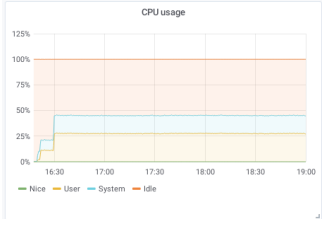


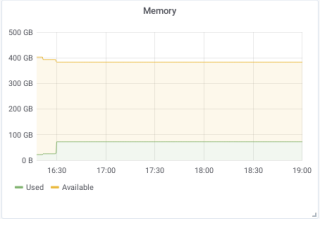
Selected integration test results

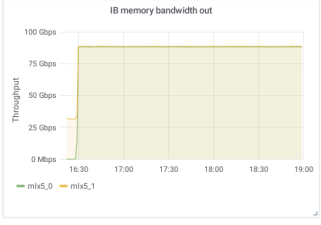












Conclusions

Upcoming challenges

- LHCb will have a **full software** trigger by 2021.
- We expect about **40**$\times$ more data, up to 40 Tbits/s.

- The entire sequence of the first trigger stage has been developed for GPUs.
- This includes decoding, clustering, tracking, Kalman filters and vertex finders.
- The amenability of GPUs to **efficiently solve** these problems has been demonstrated.

Components

- Allen is a scalable framework, with more than 70 GPU algorithms as of today.
- It is also relatively small in size (50 kLOC) and hence flexible.

- It uses a custom memory manager to avoid the cost of (de)allocating GPU memory.
- (De)allocation occurs behind the scenes $\rightarrow$ less bugs, good practice.
- ”Static” buffers require a first size calculation and a prefix sum, performed on CPU.

- The framework algorithms enforce readability.
- The framework is configurable.
- Cross-architecture support is achieved with a compile-time switch.

Integration

- Where would we place the GPUs physically?
- Would the system be compromised if GPUs are placed?
- Is it cost-effective to deploy GPUs?

Integration

- Where would we place the GPUs physically?
- Would the system be compromised if GPUs are placed?
- Is it cost-effective to deploy GPUs?

- Preferably in the event builders, which are GPU-ready.
- We have tested two integration servers and run all the expected software with headroom and no technical issues.
- Partially, the GPU cost would be amortized by the savings in network cost. The rest of the system would not have to run the first trigger stage.

Integration

- Where would we place the GPUs physically?
- Would the system be compromised if GPUs are placed?
- Is it cost-effective to deploy GPUs?

- Preferably in the event builders, which are GPU-ready.
- We have tested two integration servers and run all the expected software with headroom and no technical issues.
- Partially, the GPU cost would be amortized by the savings in network cost. The rest of the system would not have to run the first trigger stage.

GPUs have been approved, and will be implemented by 2021 in LHCb.

Acknowledgements

We would like to thank N. Neufeld and T. Colombo for many fruitful discussions. We also thank the LHCb RTA team for reviewing this work. We thank the LHCb Computing and Simulation projects for developing and maintaining the infrastructure which enabled much of the work presented. We thank the technical and administrative staff at the LHCb institutes. We acknowledge support from CERN and from the national agencies: CAPES, CNPq, FAPERJ and FINEP (Brazil); MOST and NSFC (China); CNRS/IN2P3 (France); BMBF, DFG and MPG (Germany); INFN (Italy); NWO (Netherlands); MNiSW and NCN (Poland); MEN/IFA (Romania); MSHE (Russia); MinECo (Spain); SNSF and SER (Switzerland); NASU (Ukraine); STFC (United Kingdom); DOE NP and NSF (USA). We acknowledge the computing resources that are provided by CERN, IN2P3 (France), KIT and DESY (Germany), INFN (Italy), SURF (Netherlands), PIC (Spain), GridPP (United Kingdom), RRCKI and Yandex LLC (Russia), CSCS (Switzerland), IFIN-HH (Romania), CBPF (Brazil), PL-GRID (Poland) and OSC (USA). We are indebted to the communities behind the multiple open-source software packages on which we depend. Individual groups or members have received support from AvH Foundation (Germany); EPLANET, Marie Skłodowska-Curie Actions and ERC (European Union); ANR, Labex P2IO and OCEVU, and Région Auvergne-Rhône-Alpes (France); Key Research Program of Frontier Sciences of CAS, CAS PIFI, and the Thousand Talents Program (China); RFBR, RSF and Yandex LLC (Russia); GVA, XuntaGal and GENCAT (Spain); the Royal Society and the Leverhulme Trust (United Kingdom). J. Albrecht acknowledges support of the European Research Council Starting grant PRECISION 714536.

Thanks!

Thanks a lot for listening, and get in touch if you have any questions!

dcampora@cern.ch