Allen: A High-Level Trigger on GPUs for LHCb

Citation: Computing and Software for Big Science. 2020 Apr 30;4(1):7

As Published: https://doi.org/10.1007/s41781-020-00039-7

Publisher: Springer International Publishing

Persistent URL: https://hdl.handle.net/1721.1/131603

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Computing and Software for Big Science (2020) 4:7
https://doi.org/10.1007/s41781-020-00039-7

ORIGINAL ARTICLE

Allen: A High-Level Trigger on GPUs for LHCb

R. Aaij[1] · J. Albrecht[2] · M. Belous[3,4] · P. Billoir[5] · T. Boettcher[6] · A. Brea Rodríguez[7] · D. vom Bruch[5] · D. H. Cámpora Pérez[1,8] · A. Casais Vidal[7] · D. C. Craik[6] · P. Fernandez Declara[9,10] · L. Funke[2] · V. V. Gligorov[5] · B. Jashal[11] · N. Kazeev[3,4] · D. Martínez Santos[7] · F. Pisani[9,12,13] · D. Pliushchenko[4,14] · S. Popov[3,4,15] · R. Quagliani[5] · M. Rangel[16] · F. Reiss[5] · C. Sánchez Mayordomo[11] · R. Schwemmer[9] · M. Sokoloff[17] · H. Stevens[2] · A. Ustyuzhanin[3,4,15] · X. Vilasís Cardona[18] · M. Williams[6]

Received: 18 December 2019 / Accepted: 3 April 2020

Abstract

We describe a fully GPU-based implementation of the first level trigger for the upgrade of the LHCb detector, due to start data taking in 2021. We demonstrate that our implementation, named Allen, can process the 40 Tbit/s data rate of the upgraded LHCb detector and perform a wide variety of pattern recognition tasks. These include finding the trajectories of charged particles, finding proton–proton collision points, identifying particles as hadrons or muons, and finding the displaced decay vertices of long-lived particles. We further demonstrate that Allen can be implemented in around 500 scientific or consumer GPU cards, that it is not I/O bound, and can be operated at the full LHC collision rate of 30 MHz. Allen is the first complete high-throughput GPU trigger proposed for a HEP experiment.

Keywords GPU · Real-time data selection · Trigger · LHCb

✉ R. Aaij
raaij@cern.ch

✉ D. vom Bruch
dovombru@cern.ch

✉ D. H. Cámpora Pérez
dcampora@cern.ch

[1] Nikhef National Institute for Subatomic Physics, Amsterdam, The Netherlands

[2] Fakultät Physik, Technische Universität Dortmund, Dortmund, Germany

[3] National Research University Higher School of Economics, Moscow, Russia

[4] Yandex School of Data Analysis, Moscow, Russia

[5] LPNHE, Sorbonne Université, Paris Diderot Sorbonne Paris Cité, CNRS/IN2P3, Paris, France

[6] Massachusetts Institute of Technology, Cambridge, USA

[7] Instituto Galego de Física de Altas Enerxías (IGFAE), Universidade de Santiago de Compostela, Santiago de Compostela, Spain

[8] Faculty of Science and Engineering, Maastricht University, Maastricht, The Netherlands

[9] European Organization for Nuclear Research (CERN), Geneva, Switzerland

[10] Department of Computer Science and Engineering, University Carlos III of Madrid, Madrid, Spain

[11] Instituto de Física Corpuscular, Centro Mixto Universidad de Valencia, CSIC, Valencia, Spain

[12] INFN Sezione di Bologna, Bologna, Italy

[13] Università di Bologna, Bologna, Italy

[14] National Research University Higher School of Economics, Saint Petersburg, Russia

[15] National University of Science and Technology MISIS, Moscow, Russia

[16] Instituto de Física, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

[17] University of Cincinnati, Cincinnati, OH, USA

[18] DS4DS, la Salle, Universitat Ramon Llull, Barcelona, Spain

Published online: 30 April 2020

Introduction

The LHCb detector [1] at CERN is currently being upgraded for Run 3 of the LHC. It is due to begin data taking in 2021 at an instantaneous luminosity of $\mathcal{L} = 2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, corresponding to an average of around 6 proton–proton (*pp*) collisions per LHC bunch crossing. At this luminosity, the rates of beauty and charm hadrons, which are of interest for most LHCb analyses, reach the MHz level in the LHCb detector's geometrical acceptance [2]. The majority of them decays into fully hadronic final states. Thus, efficiently reducing the output data rate requires finding charged particle trajectories (tracking) at the first level of the real-time reconstruction (trigger).

As most of LHCb's data come from its tracking detectors, which are responsible for the majority of readout channels, the upgraded detector operates a triggerless readout, in which all subdetectors are read out at the full bunch crossing rate of 30 MHz, or a maximum data rate of 40 Tbit/s. Event selection relies on two software stages. In the first stage, called HLT1, events are primarily selected using inclusive one- and two-track-based algorithms, in some cases requiring the track to be identified as a muon. At this stage, the close to optimal alignment and calibration constants from the previous run are used. HLT1 allows for an efficient reduction of the event rate by a factor 30–60, depending on the desired working point. In the second stage, called HLT2, the detector is aligned and calibrated in near-real-time and the remaining events undergo offline-quality track reconstruction, full particle identification and track fitting. Because of the high signal rate, HLT2 does not only classify bunch crossings (events) as interesting or uninteresting. Rather in most cases HLT2 identifies a decay of interest and associates it to one of the reconstructed *pp* collisions. Subsequently for most physics analyses HLT2 outputs a reduced event format one order of magnitude smaller than the raw data, consisting of only objects related to the decay of interest and the associated *pp* collision, following the approach pioneered in Run 2 [3–5]. This approach relies on the near-real-time detector alignment and calibration to maintain the ultimate detector performance without the need for costly "offline" reprocessing of the data, and results in a total output data volume of 80 Gbit/s.

Performing full track reconstruction at 30 MHz and 40 Tbit/s poses a significant computing challenge. In the baseline proposal of the upgrade data acquisition system [6, 7], data from the different LHCb subdetectors are received and combined to full events by about 250 event building x86 servers. Complete events are then sent to a separate "event filter farm" (EFF) of x86 servers, where both the HLT1 and HLT2 stages are executed. Figure 1 shows this sequence of data processing units.

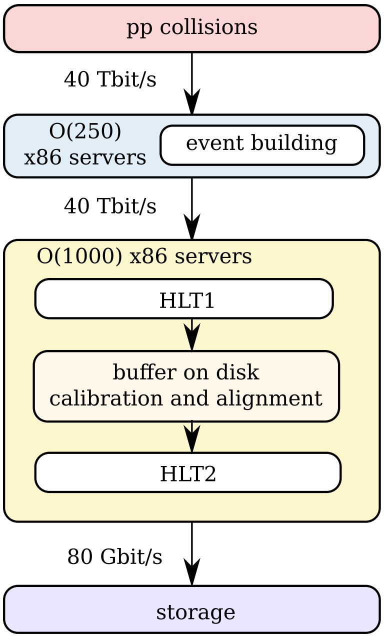


Fig. 1 In the baseline proposal for the upgraded LHCb data acquisition system, x86 event building units receive data from the subdetectors and build events by sending and receiving event fragments over a 100G Infiniband (IB) network. The full data stream of built events is sent to x86 event filter servers to process both stages of the high level trigger

As track reconstruction is an inherently parallel problem, tracking algorithms can be designed to map well to the many-core architecture of graphics processing units (GPUs). Furthermore, GPUs map well onto LHCb's data acquisition architecture, because the event building servers which host the ∼ 500 FPGA cards required to receive data from the detector at 30 MHz can also host two GPU cards each. Therefore, if the track reconstruction required for HLT1 could be processed with at most 500 GPUs, LHCb could execute HLT1 already inside the event building servers and reduce the data volume by a factor 30–60, significantly reducing the networking cost associated with sending data to the EFF.

In recent years, several particle physics experiments have studied the performance of track reconstruction on GPUs. So far only the ALICE experiment at CERN has employed GPUs in their trigger, where tracks from a single subdetector are reconstructed on the GPU, but data reduction occurs on x86 CPUs [8]. All other R&D efforts are intended for future experiments or upgrades. In most proposals, data from a single sub-detector is analyzed on the GPU at a significantly lower data rate than 40 Tbit/s [9–11]. For some, the GPU coprocessor performs track reconstruction, but event selection or data reduction occur on x86 CPUs [10]. In other cases, event selection for a single physics signature runs on the GPU [9, 11]. For Run 3, ALICE plans to perform track

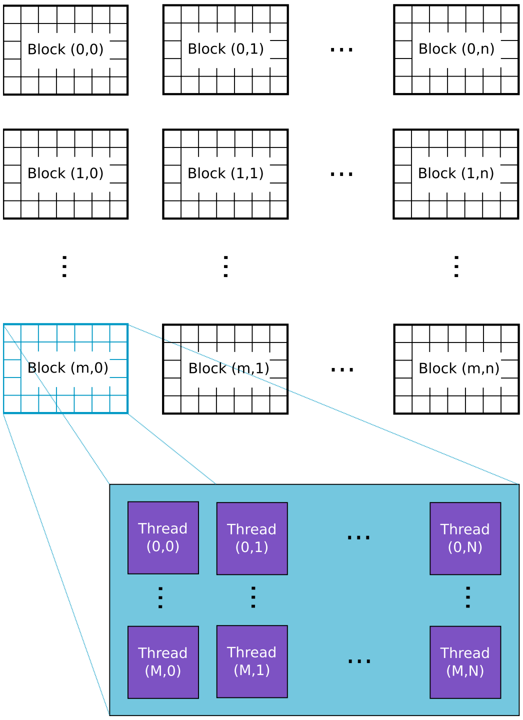


Fig. 2 Threads are grouped into blocks, forming a grid that executes one kernel on the GPU

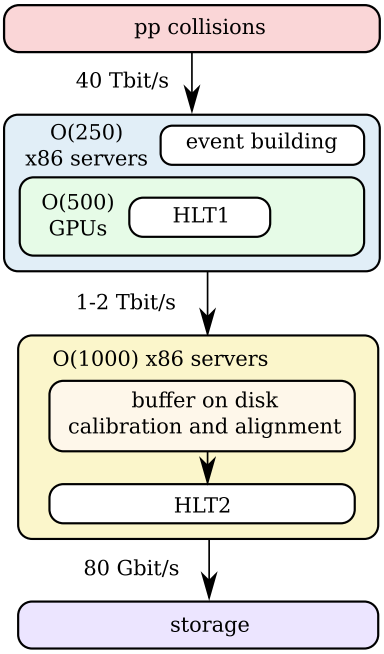


Fig. 3 In the GPU-enhanced proposal for the upgraded LHCb data acquisition system x86 event building units receive data from the subdetectors and build events by sending and receiving event fragments over a 100G Infiniband (IB) network. The same x86 servers also host GPUs which process HLT1. Only events selected by HLT1 are sent to the x86 servers processing HLT2. The data rate between the two x86 server farms is, therefore, reduced by a factor 30–60

reconstruction of more than one subdetector and data compression on the GPU, at a data rate of 5 Tbit/s [12].

In this paper, we show that for LHCb it is possible to execute a full trigger stage, including track reconstruction for several subdetectors and a variety of physics selections, at 40 Tbit/s on about 500 GPUs. We describe our implementation, named Allen after Frances E. Allen, following the LHCb convention of naming software projects after renowned scientists.

Mapping the First Trigger Stage to Graphics Processing Units

Characteristics of Graphics Processing Units

Developed for the graphics processing pipeline, GPUs excel at data parallel tasks under the SIMT paradigm [13]. An algorithm executed on the GPU is called a kernel. Every kernel is launched with many threads on the GPU executing the same instruction on different parts of the data in parallel, independently from each other. These threads are grouped into blocks within a grid, as illustrated in Fig. 2. Threads within one block share a common memory and can be synchronized, while threads from different blocks cannot communicate. The threads are mapped onto the thousands of cores available on modern GPUs for processing.

Typically, a GPU is connected to its CPU host server via a PCIe connection, which sets a limit on the bandwidth between the GPU and the CPU: 16 lanes of PCIe 3.0 and PCIe 4.0 provide 128 Gbit/s and 256 Gbit/s, respectively. From these parameters we conclude that 500 GPUs are able to consume the 40 Tbit/s data rate of the upgraded LHCb detector. The total memory on a GPU is on the order of hundreds of Gbits nowadays. Consequently, 500 GPUs should also be able to process the full HLT1 sequence if enough data processing tasks fit into GPU memory at the same time and if the tasks can be sufficiently parallelized to fully unlock the TFLOPs theoretically available on the GPU.

The Allen Concept

In our proposal, a farm of GPUs processes the full data stream, as shown in Fig. 3, which can be compared to the baseline x86-only architecture of Fig. 1. Every GPU receives complete events from an event building unit and handles several thousand events at once. Raw detector data is copied to the GPU, the full HLT1 sequence is processed on the GPU and only selection decisions and objects used for the selections, such as tracks and primary vertices, are copied

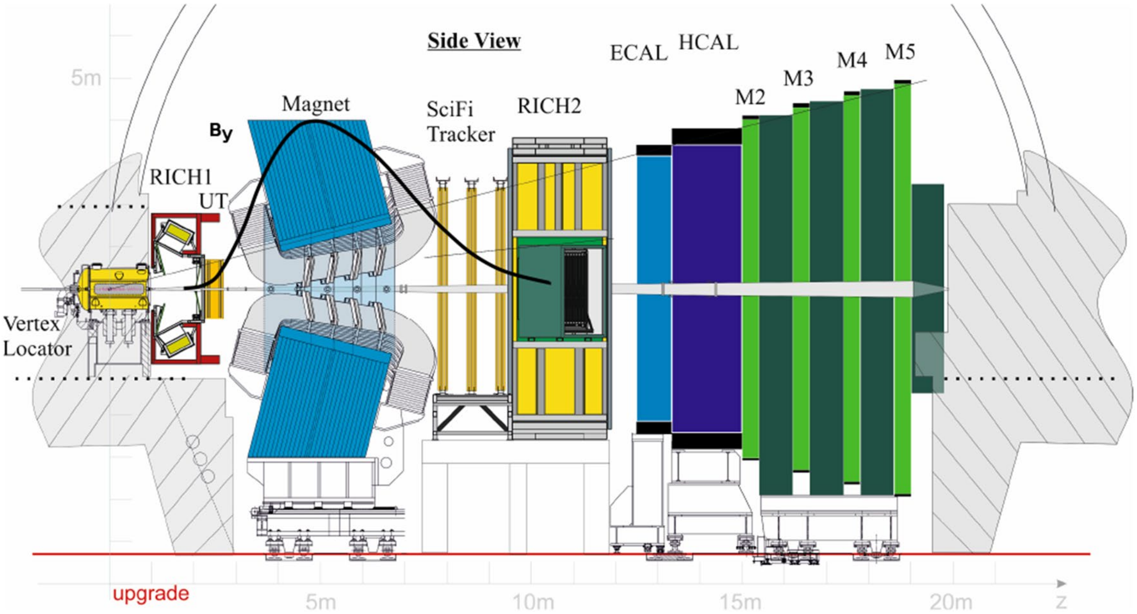


Fig. 4 Upgraded LHCb detector. The y-component of the magnetic field B_y is overlaid to visualize in which parts of the detector trajectories are bent. The maximum B_y value is 1.05 T

back to the CPU. This approach is motivated by the following considerations:

- LHCb raw events have an average size of 100 kB. When copying raw data to the GPU, the PCIe connection between the CPU and the GPU poses no limitation to the system, even when several thousand events are processed in parallel.
- Since single events are rather small, several thousand events are required to make full use of the compute power of modern GPUs.
- As the full algorithm sequence is processed on the GPU, no copies between the CPU and the GPU are required, apart from the raw input and selection output, and quantities needed to define the grid sizes of individual kernels.
- Intra-GPU communication is not required because events are independent from one another and small enough in memory footprint to be processed on a single GPU.

The project is implemented in CUDA, Nvidia's API for programming its GPUs [14]. Allen[1] includes a custom scheduler and GPU memory manager, which will be described in a companion publication.

Main Algorithms of the First Trigger Stage

A schematic of the upgraded LHCb forward spectrometer is shown in Fig. 4. The information from the tracking detectors and the muon system is required for HLT1 decisions, as described in Sect. 1. The tracking system consists of the vertex detector (Velo) [15] and the upstream tracker (UT) [16] before the magnet and tracking stations behind the magnet which are made of scintillating fibres (SciFi) [16]. The measurements from the muon detector are used to perform muon identification. The LHCb coordinate system is such that z is along the beamline, y vertical and x horizonal. The dipole magnet bends charged particle trajectories along x. Figure 4 indicates the magnitude of the y-component of the magnetic field, which extends into the UT and SciFi regions. As a consequence, tracks in the Velo detector form straight lines, while those in the UT and SciFi detectors are slightly bent.

The following recurrent tasks are performed at various stages of the HLT1 sequence:

- Decoding the raw input into coordinates in the LHCb global coordinate system.
- Clustering of measurements caused by the passage of the same particle into single coordinates ("hits"), depending on the detector type.
- Finding combinations of hits originating from the same particle trajectory (pattern recognition).
- Describing the track candidates from the pattern recognition step with a track model (track fitting).

[1] https://gitlab.cern.ch/lhcb/Allen Version 0.8 was used for the results in this publication.

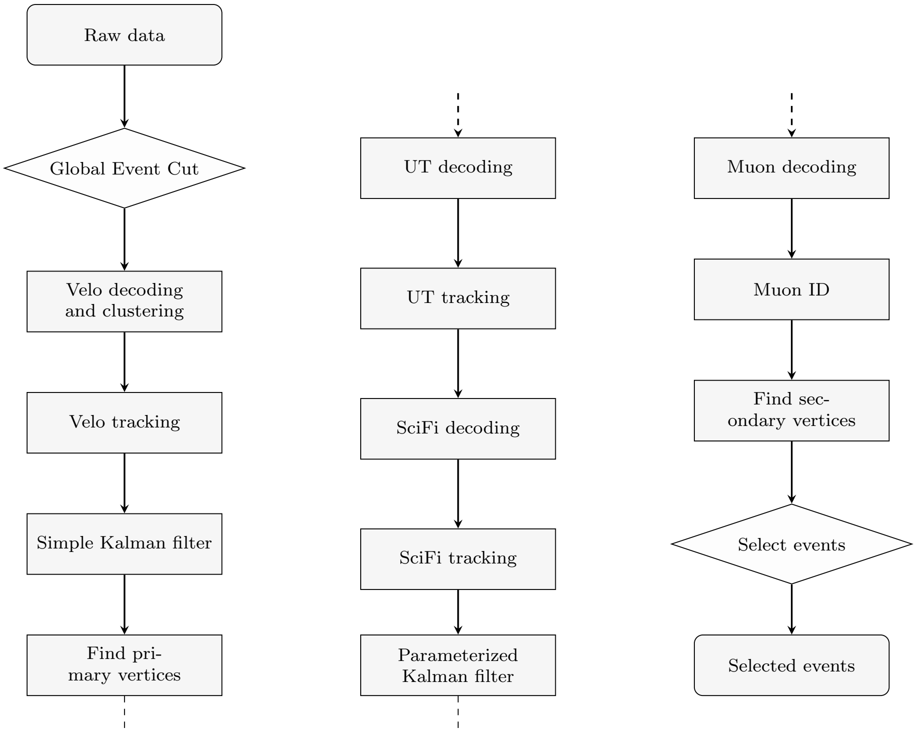


Fig. 5 Full HLT1 sequence implemented in CUDA to run on GPUs. Raw data is copied as input to the GPU, selected events are copied back to the host CPU as output. Rhombi represent algorithms reducing the event rate, while rectangles represent algorithms processing data

– Reconstructing primary and secondary vertices from the fitted tracks (vertex finding).

Figure 5 shows the full HLT1 sequence. In most cases, a single event is assigned to one block, while intra-event parallelism is mapped to the threads within one block. This ensures that communication is possible among threads processing the same event. Typically, the raw input is segmented by readout unit (for example a module of the vertex detector), so naturally the decoding can be parallelized among the readout units. During the pattern recognition step, many combinations of hits are tested and those are processed in parallel. The track fit is applied to every track and, therefore, parallelizable across tracks. Similarly, extrapolating tracks from one subdetector to the next is executed in parallel for all tracks. Finally, combinations of tracks are built when finding vertices and those can be treated in parallel.

Initially, events are preselected by a Global Event Cut (GEC) based on the size of the UT and SciFi raw data, removing the 10% busiest events. This selection is not essential for the viability of the proposed GPU architecture. It is also performed in the baseline x86 processing [7], because very busy events have a less efficient detector reconstruction and their additional physics value to LHCb is not proportionate to the computing cost of reconstructing them. The subsequent elements of the HLT1 sequence are now described in turn.

Velo Detector

The Velo detector consists of 26 planes of silicon pixel sensors placed around the interaction region. Its main purpose lies in reconstructing the *pp* collisions (primary vertices or PVs) and in creating seed tracks to be further propagated through the other LHCb detectors. The Velo track reconstruction is fully described in an earlier publication [17] and is recapped here for convenience.

The reconstruction begins by grouping measurements caused by the passage of a particle within each silicon plane into clusters, an example of a more general process known as connected component labeling. Allen uses a clustering algorithm employing bit masks, which searches for clusters locally in small regions. Every region can be treated independently, allowing for parallel processing.

Straight-line tracks are reconstructed by first forming seeds of three hits from consecutive layers (“triplets”), and then extending these to the other layers in parallel. We exploit the fact that prompt particles produced in *pp* collisions traverse the detector in lines of constant ϕ angle (within a cylindrical coordinate system where the cylinder axis coincides with the LHC beamline) and sort hits on every layer by ϕ for fast look-up when combining hits to tracks.

Velo tracks are fitted with a simple Kalman filter [18] assuming that the x- and y-components are independent from one another and assigning a constant average transverse momentum of 400 MeV to all tracks for the noise contribution from multiple scattering.

Finally, we search for PVs in a histogram of the point of closest approach of tracks to the beamline, where a cluster indicates a PV candidate. We refrain from a one-to-one mapping between a track and a vertex, which would introduce dependencies between the fitting of individual vertex candidates and would require sequential processing. Instead, every track is assigned to every vertex based on a weight, so that all candidates can be fitted in parallel.

UT Detector

Four layers of silicon strip detectors make up the UT detector, the strips of the two outer layers are aligned vertically, the two inner layers are tilted by +5° and −5° around the z-axis, respectively. Since more than 75% of the hits consist of only one fired strip, no clustering is performed in this subdetector. The UT hits are decoded into regions based on their x-coordinate. Every region is then sorted by the y-coordinate. This allows for a fast look-up of hits around the position of an extrapolated Velo track. Velo tracks are extrapolated to the UT detector based on a minimum momentum cut-off of 3 GeV, resulting in a maximal bending allowed between the Velo and UT detectors.There is no requirement on the transverse momentum. Subsequently, UT hits are assigned to Velo tracks and the track momentum is determined from the bending between the Velo and UT fitted straight-line track segments with a resolution of about 20%. The UT decoding and tracking algorithms are described in more detail in Ref. [19].

SciFi Detector

The SciFi detector consists of three stations with four layers of scintillating fibres each, where the four layers of every station are in x–u–v–x configuration. The u- and v-layers are tilted by +5° and −5°, respectively, while the x-layers are vertical. The clustering of the SciFi hits and sorting along x is performed on the readout board; therefore, sorted clusters are obtained directly when decoding.

Tracks passing through both the Velo and UT detectors are extrapolated to the SciFi detector using a parameterization based on the track direction and the momentum estimate obtained after the UT tracking. This avoids loading the large magnetic field map into GPU memory. A search window defined by the UT track properties and a maximum number of allowed hits is determined for every UT track and every SciFi layer.

The hit efficiency of the scintillating fibres is 98–99%; therefore, several seeds are allowed per UT track, so that the track reconstruction efficiency is not limited by requiring hits from specific layers. Seeds are formed combining triplets of hits from within the search windows of one x-layer in each of the three SciFi stations. The curvature of tracks inside the SciFi region due to the residual magnetic field tails from the LHCb dipole is taken into account when selecting the best seeds. Only the seeds with the lowest χ^2 relative to a parameterized description of the track within the SciFi volume are then extended by adding hits from the remaining x-layers, using the same track description. Since only the information of three hits is used for the χ^2, its discriminating power is limited. Therefore, multiple track seeds are processed per UT track.

The magnetic field inside the SciFi detector can be expressed as $B_y(z) = B_0 + B_1 \cdot z$ and it is found that at first order $\frac{B_1}{B_0}$ is a constant. Using this parameterization, tracks are projected onto the remaining x and u/v-layers, and hits that deviate the least from the reference trajectory, within a track-dependent acceptance, are added. Only the U/V-layers provide information on the track motion in the y–z plane. Thus, a parameterization accounting for the small curvature in the y–z plane is also taken into account in the track model, once all hits have been added.

Finally, a least means square fit is performed both in x and y. Every track is assigned a weight based on the normalized x-fit χ^2, y-fit χ^2, and the number of hits in the track. Only the best track is accepted per UT track, reducing fake tracks as much as possible.

Muon Detector

The muon system [20] consists of four multiwire proportional chambers interleaved with iron walls. Every station is divided into four regions with chambers of different granularity. Hits are read out with pads and strips, while strips from the same station can overlap to give a more accurate position measurement. During the decoding of muon measurements, such crossing strips are combined into a single hit. For muon identification, the "isMuon" algorithm described in Ref. [21] is employed: tracks are extrapolated from the SciFi to the muon stations and muon hits are matched to a track within a region defined by the track properties. Depending on the track momentum, hits in different numbers of stations are required for a track to be tagged as a muon.

Kalman Filter

A Kalman filter is applied to all tracks to improve the impact parameter resolution, where the impact parameter (IP) is the distance between the point of closest approach of a track and a PV. The nominal LHCb Kalman filter uses a Runge-Kutta extrapolator to propagate track states between measurements and a detailed detector description to determine noise due to multiple scattering. In order to increase throughput and limit memory overhead, these costly calculations are replaced with parameterizations. Two versions of the parameterized Kalman filter are implemented in Allen: one which takes into account the whole detector and one which fits only the Velo track segment but using the estimated momentum from the full track passing through the Velo, UT and SciFi detectors. Since the impact parameter is mainly influenced by the measurements nearest to the interaction region, the Velo-only Kalman filter is used in the HLT1 sequence. This results in a significant computing speedup compared to applying the full Kalman filter.

Selections

Given the momentum, impact parameter and position information from the track fit as well as the muon identification, selections are applied on single tracks and two-track vertices similarly to the HLT1 selections used in Run 2 of LHCb [22–24]. Secondary vertices are fitted in parallel from combinations of two tracks each, providing a momentum and mass estimate for the hypothetical decaying particle, assigning the pion mass hypothesis to all tracks except for those identified as muons, for which the muon mass is assigned. The following five selection algorithms, which cover the majority of the LHCb physics programme and which are similar to lines accounting for about 95% of the HLT1 trigger rate in Run 2 [22], are implemented in Allen:

- 1-Track: A single displaced track with $p_{\mathrm{T}} > 1\,\mathrm{GeV}$.
- 2-Track: A two-track vertex with significant displacement and $pt > 700$ MeV for both tracks.
- High-p_T muon: A single muon with $p_{\mathrm{T}} > 10$ GeV for electroweak physics.
- Displaced dimuon: A displaced dimuon vertex with $p_{\mathrm{T}} > 500$ MeV for both tracks.
- High-mass dimuon: A dimuon vertex with mass near or larger than the J/Ψ mass with $p_{\mathrm{T}} > 750$ MeV for both tracks.

Results

The performance of Allen is studied both with respect to the computing throughput per GPU and the physics outcome in terms of track reconstruction efficiency and event selection efficiency for various representative LHCb analyses.

Physics Performance

For physics studies, simulated samples enhanced with decay channels of interest for the LHCb physics program are employed, namely a combination of 5000 events of each of the following decays: $B^0 \to K^{*0}\mu^+\mu^-$, $B^0 \to K^{*0}e^+e^-$, $B^0_s \to \phi\phi$, $D^+_s \to K^+K^-\pi^+$ and $Z \to \mu^+\mu^-$. Efficiencies of track and vertex reconstruction, muon identification and trigger selections, as well as the momentum resolution are determined directly within the Allen framework.

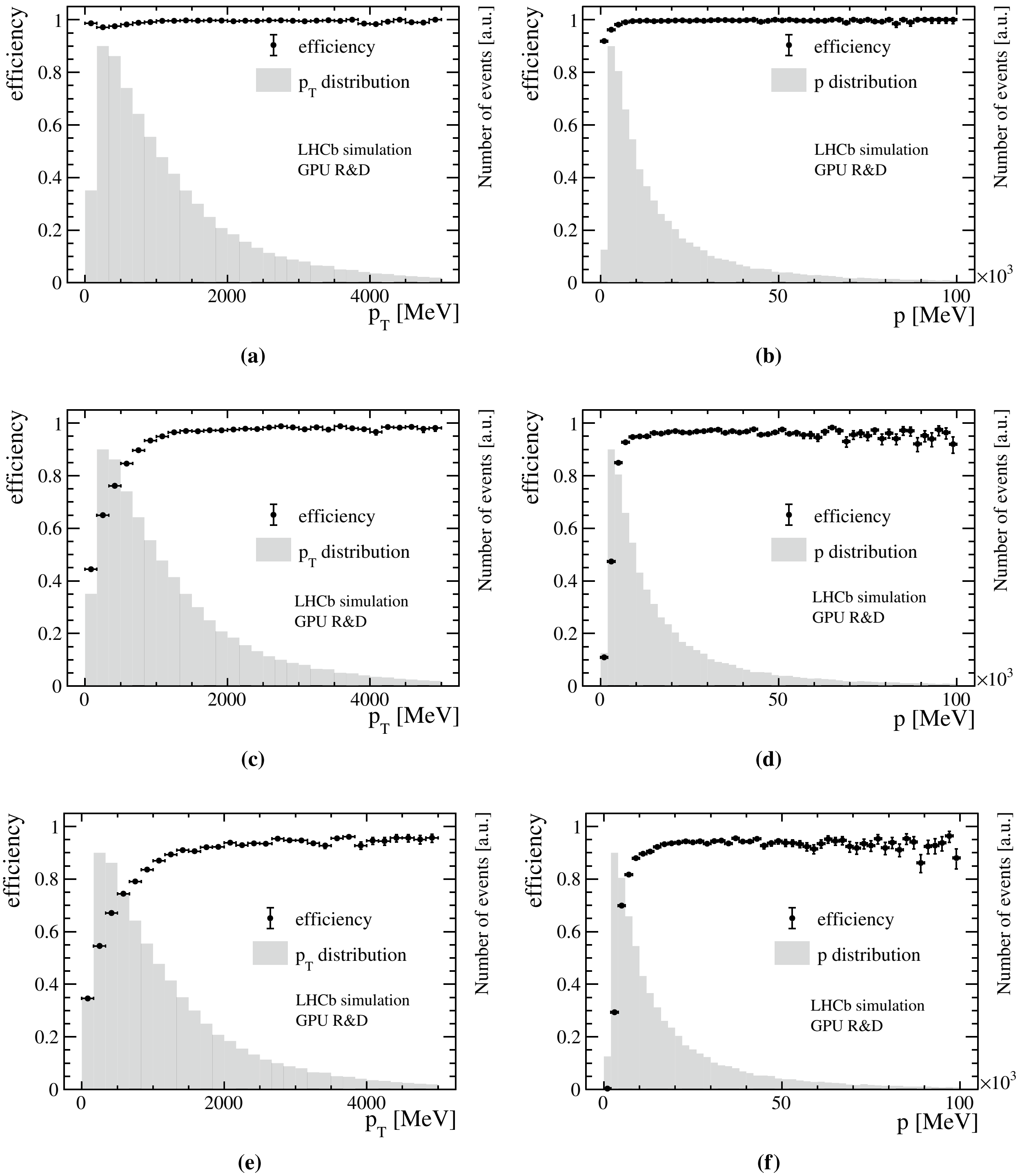
In LHCb, tracks are defined as correctly reconstructed if at least 70% of the hits match those of the Monte Carlo (MC) particle associated to the track in simulation. Only MC particles resulting in the following minimum numbers of hits are considered as “reconstructible tracks”: at least one hit in at least three different Velo modules and at least one hit in an x- and a u/v-layer in the UT detector and every station in the SciFi detector. Figure 6 shows the track reconstruction efficiency of correctly reconstructed tracks in the Velo (top), Velo and UT (middle), Velo, UT and SciFi (bottom) detectors versus transverse momentum p_{T} and momentum p with respect to reconstructible tracks originating from B decays. A reconstructed PV is matched to a simulated PV if the distance is less than five times the uncertainty of the reconstructed PV along the z-axis. Figure 7 shows the reconstruction efficiency of PVs versus the track multiplicity of the MC PV. As displayed in Fig. 8, a relative momentum resolution better than 1 % is achieved which is sufficient for the selections of HLT1 and can be compared to a resolution of 0.5–1% obtained from offline-quality track reconstruction during Run 2. The muon identification efficiency is shown in Fig. 9. It is determined with respect to “reconstructible muons”, defined as reconstructed tracks which were matched to a muon MC particle.

Finally, the trigger rates for the five selections are shown in Table 1. The total HLT1 output rate is about 1 MHz, therefore, reducing the event rate by a factor 30. For this output rate, the selection efficiencies for various decay channels are given in Table 2. We quote the efficiency of the GEC, as well as for “TIS” events, with at least one passing trigger candidate not associated with a true signal decay product, and for “TOS” events, where the signal decay products must pass the trigger selection themselves.

Figure 10 illustrates the difference in efficiency and rate for the 1-Track and 2-Track trigger lines for the $B^0_s \to \phi\phi$ sample between fitting tracks with the simple Kalman filter versus the parameterized Kalman filter, when varying the selection criteria of the IP χ^2. The IP χ^2 is defined as the difference between the χ^2 of the PV reconstructed with and without the track under consideration and serves as estimate for the track displacement. Especially the efficiency of the 2-Track line improves when using the parameterized Kalman filter, since the momentum threshold for individual tracks is lower compared to the 1-Track line.

Computing Performance

For throughput studies, simulated samples of minimum bias events are used, representing the physics conditions



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Fig. 6 Track reconstruction efficiency versus transverse momentum p_T (left) and momentum p (right) of reconstructed tracks passing through the Velo (**a**, **b**), Velo and UT (**c**, **d**), Velo, UT and SciFi detectors (**e**, **f**) with respect to reconstructible non-electron tracks passing through the Velo, UT and SciFi detectors and produced from B decays within the pseudorapidity coverage of the LHCb detector, $2 < \eta < 5$, for all signal samples combined. The p_T and p distributions are overlaid as histograms

expected for Run 3. The computing performance is compared among different Nvidia GPU cards. In all cases, Allen is compiled with gcc 8.2 [25] and CUDA 10.1. The HLT1 sequence is run on a configurable number of concurrent threads. Each thread employs a GPU stream to asynchronously execute kernels and perform data transmission between CPU and GPU, such that memory transmissions do not impact throughput. The timer is started

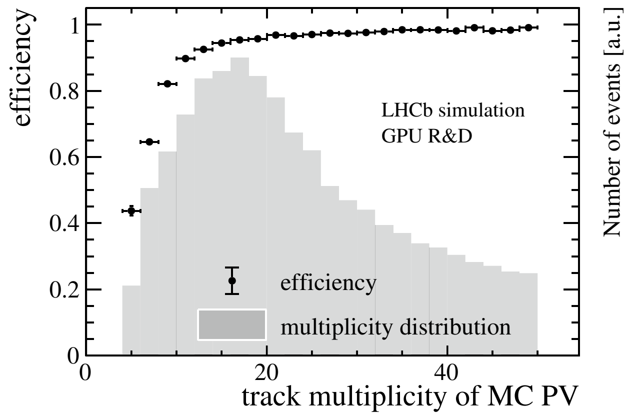


Fig. 7 PV reconstruction efficiency versus track multiplicity of the MC PV for minimum bias events. The track multiplicity distribution is overlaid as a histogram

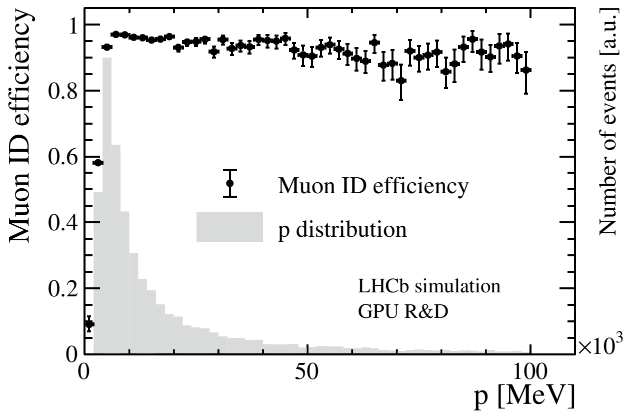


Fig. 9 Muon identification efficiency versus momentum for tracks passing through the Velo, UT and SciFi detectors with respect to all reconstructible muons (explained in the text), for all signal samples combined. The momentum distribution is overlaid as a histogram

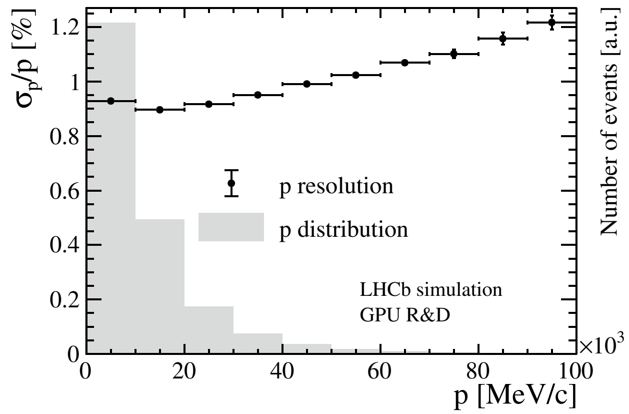


Fig. 8 Relative momentum resolution of tracks passing through the Velo, UT and SciFi detectors versus momentum for all signal samples combined. Points represent the mean, error bars the width of a Gaussian distribution fitted to the resolution in every momentum slice. The momentum distribution is overlaid as a histogram

Table 1 Rates of the five trigger selections implemented in Allen and the total HLT1 output rate, determined with minimum bias events

Trigger	Rate [kHz]
1-Track	215 ± 18
2-Track	659 ± 31
High-p_T muon	5 ± 3
Displaced dimuon	74 ± 10
High-mass dimuon	134 ± 14
Total	999 ± 38

Table 2 Efficiencies of the total HLT1 selection. The TIS -OR- TOS and TOS efficiencies are calculated using events passing the GEC (definitions for TIS, TOS and GEC are in the text). All efficiencies and their uncertainties are quoted in percentages and are determined from the different signal samples, with selections resulting in the rates given in Table 1. Signal events are selected with the following criteria: b and c hadrons have a $p_{\mathrm{T}} > 2$ GeV and a lifetime $\tau > 0.2$ ps. Children of b and c hadrons have $p_{\mathrm{T}} > 200$ MeV. Children of Z bosons have $p_{\mathrm{T}} > 20$ GeV

Signal	GEC	TIS -OR- TOS	TOS	GEC × TOS
$B^0 \to K^{*0}\mu^+\mu^-$	89 ± 2	91 ± 2	89 ± 2	79 ± 3
$B^0 \to K^{*0}e^+e^-$	84 ± 3	69 ± 4	62 ± 4	52 ± 4
$B^0_s \to \phi\phi$	83 ± 3	76 ± 3	69 ± 3	57 ± 3
$D^+_s \to K^+K^-\pi^+$	82 ± 4	59 ± 5	43 ± 5	35 ± 4
$Z \to \mu^+\mu^-$	78 ± 1	99 ± 0	99 ± 0	77 ± 1

prior to processing a sequence in the first stream, and it is stopped after all streams have returned.

For most measurements, 12 thread-stream pairs with 1000 events each were processed 100 times, allocating 700 MB of GPU memory for every stream. Only in the case of the GTX 670, GTX 680 and the GTX 1060 6GB two thread-stream pairs were used instead. The measurement was performed 10 times with different sets of 1000 events each. The mean and standard deviation of the 10 measurements are shown for various Nvidia GPU cards as a function of their theoretical peak 32-bit FLOPS performance in Fig. 11. The minimum rate per GPU necessary for processing the 30 MHz input rate with 500 GPUs is 60 kHz. Three cards surpass this threshold with a margin, namely the RTX 2080 Ti, the V100 and the Quadro RTX 6000, currently the best cards in the consumer, scientific and professional lines of Nvidia, respectively. Analyzing the performance as a function of theoretical peak 32-bit FLOPS performance reveals how the application scales to the hardware under study. The linear dependence visible in Fig. 11 shows that the Allen code makes efficient use of the computing architecture and is likely to scale well to future generations of GPU processors.

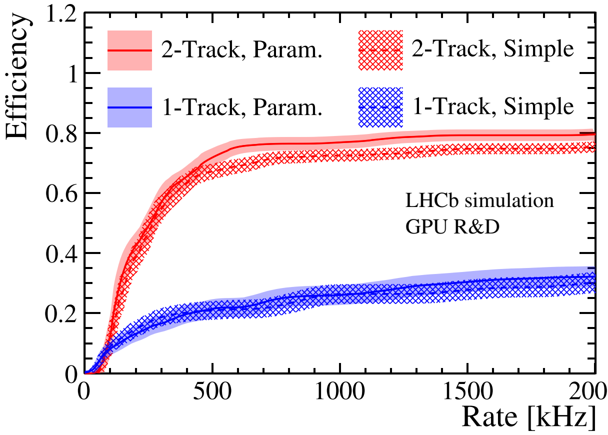


Fig. 10 Efficiency of the 1-Track and 2-Track trigger lines when calculating the IP χ^2 (see text for definition) from tracks fitted with the simple and parameterized Kalman filter, using the $B_s^0 \to \phi\phi$ sample. Varying the selection criteria of the IP χ^2 results in rate and efficiency changes. The efficiency is calculated from subsets of the sample, the central value and error band correspond to the mean and standard deviation, respectively

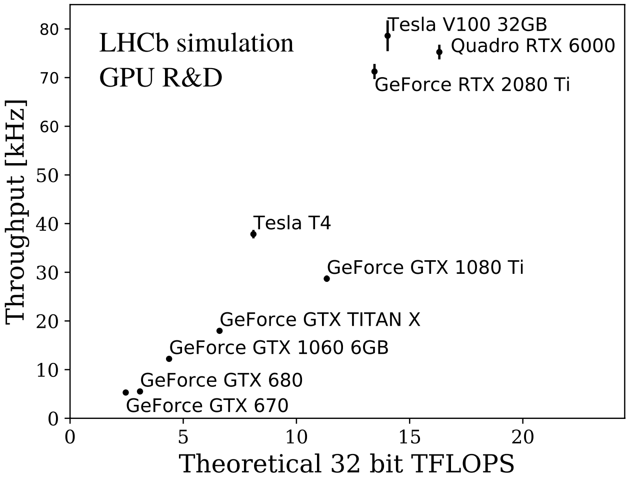


Fig. 11 Allen throughput on various GPUs with respect to their reported peak 32-bit FLOPS performance. The mean and standard deviation of 10 measurements with different sets of 1000 events each are shown in the figure, with the measurement setup as described in the text

The throughput as a function of the occupancy in the SciFi detector is depicted in Fig. 12. The slower throughput decrease in the high occupancy region gives confidence that Allen can be adapted to real data taking conditions, where the detector occupancy might be higher than in simulation (as observed consistently during Runs 1 and 2). If the GEC removing the 10% busiest events is deactivated and all events are processed, the Allen throughput drops by about 20%.

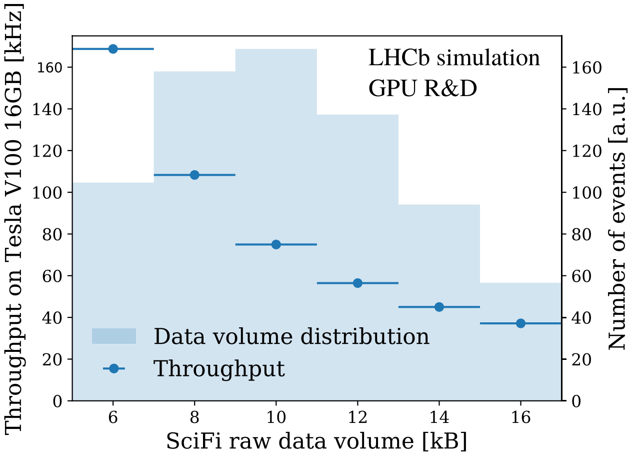


Fig. 12 Throughput of the Allen sequence as a function of the SciFi raw data volume, which is proportional to the SciFi occupancy. The measurement setup is described in the text. For every data point, 1000 different events within the range of the SciFi raw data volume bin are processed. The GEC removing the 10% busiest events was deactivated for these measurements

Conclusions

We present Allen, an implementation of the first trigger stage of LHCb for Run 3 entirely on GPUs. This is the first complete high-throughput GPU trigger proposed for a HEP experiment. Allen covers the majority of the LHCb physics programme, using an analogous reconstruction and selection sequence as in Run 2. The demonstrated event throughput shows that the full HLT1 sequence can run on about 500 of either one of the RTX 2080 Ti, V100 or Quadro RTX 6000 Nvidia GPU cards. Consequently, the GPUs can be hosted by the event building servers, significantly reducing the network cost associated with sending HLT1 output to the EFF. We show that the performance in terms of track and vertex reconstruction efficiencies, muon identification and momentum resolution are sufficient for efficient trigger selections for analyses representative of the LHCb physics programme.

Acknowledgements We would like to thank N. Neufeld and T. Colombo for many fruitful discussions. We also thank the LHCb RTA team for supporting this publication and reviewing this work. 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 (The Netherlands); MNiSW and NCN (Poland); MEN/IFA (Romania); MSHE (Russia); MinECo (Spain); SNSF and SER (Switzerland); NASU (Ukraine); STFC (UK); DOE NP and NSF (USA). We acknowledge the computing resources that are provided by CERN, IN2P3 (France), KIT and DESY (Germany), INFN (Italy), SURF (The Netherlands), PIC (Spain), GridPP (UK), 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 (UK). J. Albrecht acknowledges support of the European Research Council Starting Grant PRECISION 714536. D. vom Bruch, V. V. Gligorov, F. Reiss and R. Quagliani acknowledge support of the European Research Council Consolidator Grant RECEPT 724777. H. Stevens, L. Funke acknowledge support of the Collaborative Research Center SFB 876. T. Boettcher and M. Williams are supported by US NSF Grant PHY-1912836. D. Craik is supported by US NSF Grants OAC-1836650 and PHY-1904160. A. Ustyuzhanin is supported by the Russian Science Foundation Grant Agreement No. 19-71-30020. D. Martínez Santos and A. Brea Rodríguez acknowledge support from the European Research Council Starting Grant BSMFLEET 639068.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. LHCb collaboration, Aaij R, et al. (2015) LHCb detector performance. Int J Mod Phys A30: 1530022
2. Fitzpatrick C, Gligorov VV (2014) Anatomy of an upgrade event in the upgrade era, and implications for the lhcb trigger. Technical report LHCb-PUB-2014-027. CERN-LHCb-PUB-2014-027, CERN, Geneva
3. Aaij R et al (2016) Tesla: an application for real-time data analysis in high energy physics. Comput Phys Commun 208:35
4. Aaij R et al (2019) A comprehensive real-time analysis model at the LHCb experiment. JINST 14:P04006
5. LHCb Collaboration (2018) Computing Model of the Upgrade LHCb experiment. Technical report. CERN-LHCC-2018-014. LHCB-TDR-018, CERN, Geneva
6. Albrecht J, et al. (2018) Upgrade trigger and reconstruction strategy: 2017 milestone. Technical report. LHCb-PUB-2018-005. CERN-LHCb-PUB-2018-005, CERN, Geneva
7. LHCb Collaboration (2018) LHCb trigger and online technical design report. Technical report. CERN-LHCC-2014-016, CERN, Geneva
8. Rohr D et al (2012) ALICE HLT TPC tracking of pb-pb events on GPUs. J Phys Conf Ser 396:012044
9. Sen P, Singhal V (2015) Event selection for MUCH of CBM experiment using GPU computing. In: 2015 annual IEEE India Conference (INDICON). IEEE, New Delhi, 2015, pp 1–5, 17–20 Dec 2015
10. Funke D et al (2014) Parallel track reconstruction in CMS using the cellular automaton approach. J Phys Conf Ser 513:052010
11. vom Bruch D (2017) Online data reduction using track and vertex reconstruction on GPUs for the Mu3e experiment. EPJ Web Conf 150:00013
12. Rohr D, Gorbunov S, Lindenstruth V (2017) GPU-accelerated track reconstruction in the ALICE high level trigger. J Phys Conf Ser 898:032030
13. Lindholm E, Nickolls J, Oberman S, Montrym J (2008) NVIDIA Tesla: a unified graphics and computing architecture. IEEE Micro 28(2):39–55
14. CUDA Toolkit. https://docs.nvidia.com/cuda/. Accessed 11 Feb 2020
15. LHCb collaboration (2013) LHCb VELO upgrade technical design report. Technical report. CERN-LHCC-2013-021, CERN, Geneva
16. LHCb collaboration (2014) LHCb tracker upgrade technical design report. Technical report. CERN-LHCC-2014-001, CERN, Geneva
17. Cámpora Pérez DH, Neufeld N, Riscos Nuñez A (2019) A fast local algorithm for track reconstruction on parallel architectures. In: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, pp 698–707
18. Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng 82:35
19. Fernandez Declara P et al (2019) A parallel-computing algorithm for high-energy physics particle tracking and decoding using gpu architectures. IEEE Access 7:91612
20. LHCb collaboration (2013) LHCb PID upgrade technical design report Technical report. CERN-LHCC-2013-022, CERN, Geneva
21. Archilli F et al (2013) Performance of the muon identification at LHCb. JINST 8:P10020
22. Aaij R et al (2018) Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. JINST 14:P04013
23. Likhomanenko T et al (2015) LHCb Topological trigger reoptimization. J Phys Conf Ser 664:082025
24. Gligorov VV, Williams M (2013) Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree. JINST 8:P02013
25. GCC, the Gnu Compiler Collection. https://gcc.gnu.org. Accessed 11 Feb 2020