Proceedings of CTD 2020
PROC-CTD2020-50
June 26, 2020

Allen: A High Level Trigger on GPUs for LHCb

Thomas Boettcher

On behalf of the LHCb Real Time Analysis project,
Department of Physics
Massachusetts Institute of Technology, United States

ABSTRACT

The upgraded LHCb detector will begin taking data in 2021 with a triggerless readout system. As a result the full 30 MHz inelastic event rate will be processed using a software-only High Level Trigger (HLT). This will allow for dramatic improvements in LHCb's ability to study beauty and charm hadron decays, but also presents an extraordinary technical challenge and has prompted the study of alternative hardware technologies. In this talk I will discuss the Allen project, a framework for implementing LHCb's first stage HLT (HLT1) on GPUs. I will focus on the development and performance of the full HLT1 reconstruction sequence executed on GPUs, including reconstruction algorithms developed and optimized specifically for many-core architectures.

PRESENTED AT

Connecting the Dots Workshop (CTD 2020)
April 20-30, 2020

1 Introduction

After the ongoing long shutdown, the LHC will begin providing proton-proton collisions at an upgraded center of mass energy of 14 TeV, and LHCb will take data at a luminosity of $\mathcal{L} = 2 \times 10^{33} \text{cm}^{-2}\text{s}^{-1}$. Figure 1 shows production rates for some standard model particles at the upgraded LHC. ATLAS and CMS primarily study processes such as production of Higgs bosons, top quarks, and electroweak bosons. At the upgraded LHC's design energy and luminosity, these processes will occur at rates of a few kHz or less. These processes also produce high-energy particles. As a result, ATLAS and CMS can effectively trigger on these processes at rates of less than about 100 kHz using high-energy signatures in single detector systems, such as high-E_{T} calorimeter clusters.

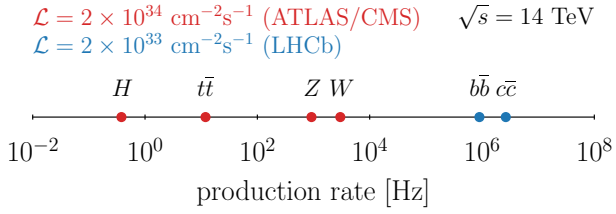


Figure 1: Event rates for various interesting standard model processes at the upgraded LHC's design energy and expected instantaneous luminosities. Calculated using MadGraph5_aMC@NLO [1].

The LHCb experiment's primary goal is to study the decays of hadrons containing b or c quarks. The combined production rate of these heavy flavor hadrons will exceed 1 MHz in the LHCb detector's acceptance [2]. Furthermore, these hadrons will decay to final state particles with $p_{\text{T}} \lesssim 1$ GeV, which is comparable to other particles produced in the underlying pp event. Figure 2 shows the expected Run 3 performance of an algorithm similar to those used in the LHCb hardware trigger to select hadronic heavy flavor decays in Runs 1 and 2. This algorithm selects events based only on the highest energy 2x2 cell cluster in the hadronic calorimeter. In Runs 1 and 2, the LHCb hardware trigger operated at an output rate of 1 MHz, which would result in trigger efficiencies of around 10% in Run 3.

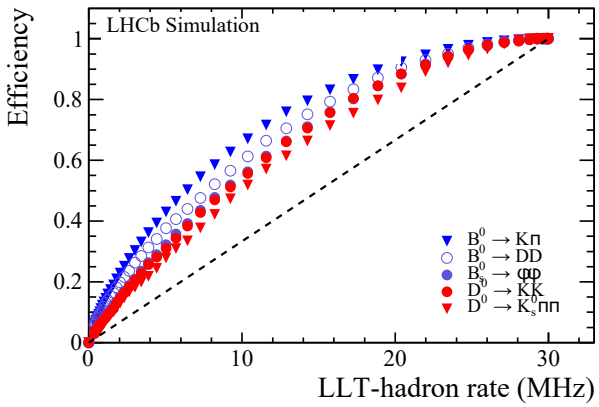


Figure 2: Efficiency versus trigger rate for various heavy flavor decays using a hardware trigger algorithm that selects events based on the highest E_{T} calorimeter cluster in the event. The dashed line shows the performance of randomly selecting events. From Ref. [4].

Instead of relying on high-energy signatures, LHCb triggers on heavy flavor decays using displaced tracks and secondary vertices. This strategy requires combining information from the entire tracking system. As a result, LHCb will operate without a hardware trigger in Run 3. The entire detector will be read out at the

LHC bunch crossing rate of 40 MHz and processed using software triggers [4]. This presents a significant computing challenge and has resulted in efforts to redesign LHCb's high level trigger.

I will present the Allen project, a software trigger on GPUs for Run 3 at LHCb. I will summarize the Allen framework and the algorithms that make up LHCb's first level software trigger (HLT1) sequence. I will also discuss Allen's physics performance and throughput. Finally I will discuss ongoing work and the ways Allen could allow LHCb to expand its physics program in Run 3. Results shown here are based on Allen version 0.8 documented in Ref. [3]*.

2 The Allen Framework

The Allen project is named after Frances E. Allen and began as a GPU research and development project in February 2018. Allen is a standalone application requiring only CUDA v10.2 and a C++17 compatible compiler. Allen includes a framework for complex GPU workflows, as well as a sequence of algorithms performing LHCb's entire HLT1 reconstruction and selection sequence. After an extensive review process, Allen was recently chosen as LHCb's baseline HLT1 for Run 3.

Allen is based on a framework that effectively hides some difficult aspects of general purpose GPU programming and allows developers to focus on algorithm development. Figure 3 shows an overview of this framework. A custom memory manager and scheduler handle allocating and deallocating GPU memory based on each algorithm's data dependencies. Allen can also be compiled for CPUs, so new developers can contribute without access to GPUs. This accessibility has allowed Allen to grow under a development team consisting primarily of students, including undergraduate and masters students. It is also necessary for allowing the Allen developer base to grow, as most LHCb collaboration members have little GPU programming experience.

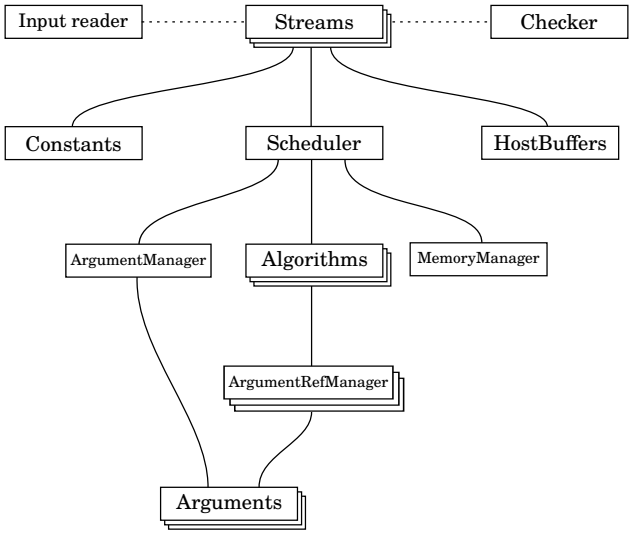


Figure 3: An overview of the Allen framework. From Ref. [5].

Only the algorithms implemented in Allen are specific to the LHCb experiment. Allen could easily host non-LHCb algorithms. This would allow Allen to serve as a platform for other high-throughput GPU applications.

*Source code is available at `https://gitlab.cern.ch/lhcb/Allen/-/releases/v0.8`

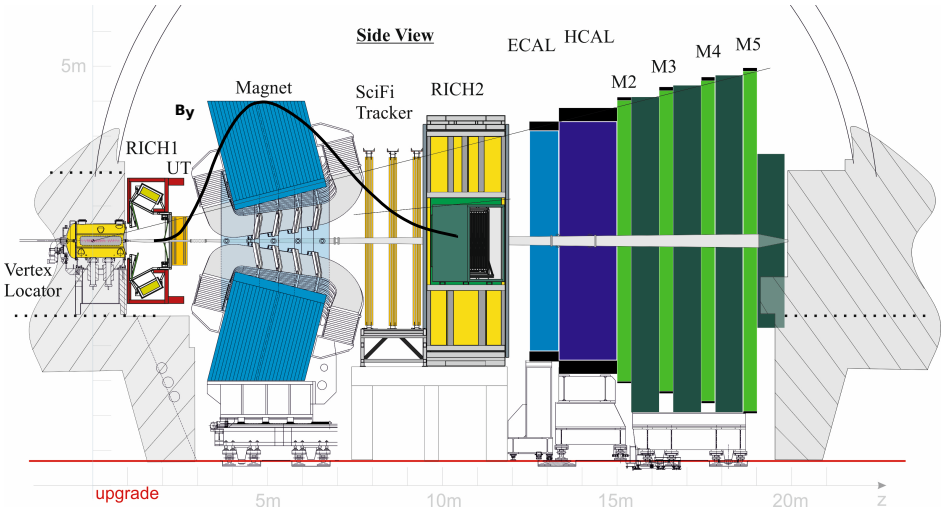


Figure 4: The upgraded LHCb detector. A sketch of the relative magnetic field strength is overlaid in black. From Ref. [3].

3 The Allen HLT1 Sequence

The Allen HLT1 sequence performs a partial reconstruction of charged tracks. Figure 4 shows the upgraded LHCb detector. Allen reconstructs tracks using the VELO, UT, and SciFi subdetectors. The sequence also reconstructs primary vertices and performs muon identification. Charged tracks are fit using a Kalman filter and combined to form two-track secondary vertices. Finally events are selected based on the reconstructed tracks and secondary vertices.

3.1 VELO Tracking

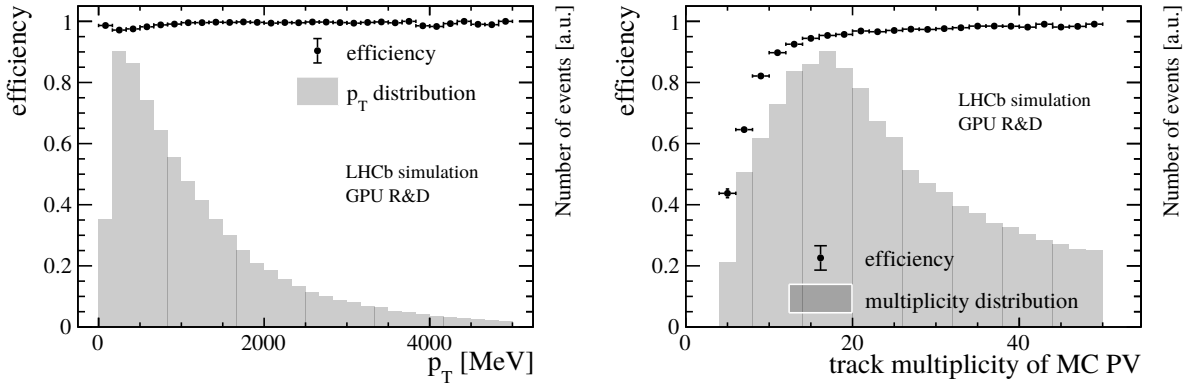
The LHCb Vertex Locator (VELO) is made up of 26 layers of silicon pixel detectors. During stable beams the VELO closes around the beamline, resulting in a minimum distance of 5.1 mm between the beamline and the sensitive region of the detector. The VELO provides tracks used for reconstructing primary vertices and is crucial for determining the displacement of reconstructed tracks and secondary vertices.

VELO pixels are clustered in constant time using a bit mask clustering algorithm. Seed candidates consist of activated pixels surrounded by unactivated pixels to their north, northeast, east, and southeast. Activated pixels among the eight surrounding pixels are added to the cluster. This is then repeated with newly added pixels until no additional pixels can be added to the cluster.

The VELO track finding algorithm is described in Ref. [6]. Charged particles originating from the beamline will produce hits of constant azimuthal angle ϕ. To take advantage of this, clusters are sorted in ϕ, and three-hit triplets are created from clusters falling within a ϕ search window. Triplets are then forwarded to the next VELO layer where another hit can be added to the track. The triplet creation and forwarding is repeated until every layer has been processed. At every layer, each cluster is processed in parallel. Figure 5a shows the VELO tracking efficiency.

3.2 Primary Vertex Finding

Reconstructed VELO tracks are used to locate primary vertices. These tracks are extrapolated to their closest point to the beamline and a histogram is filled using a Gaussian kernel based on their z positions and uncertainties. This is performed in parallel for each track. Primary vertex seeds are created from peaks in this histogram and their positions are determined using a weighted minimum χ^2 fit. Figure 5b shows the PV finding efficiency.



(a) (b)

Figure 5: VELO tracking efficiency (a) and PV finding efficiency (b). Generated distributions are shown in the shaded regions. From Ref. [3].

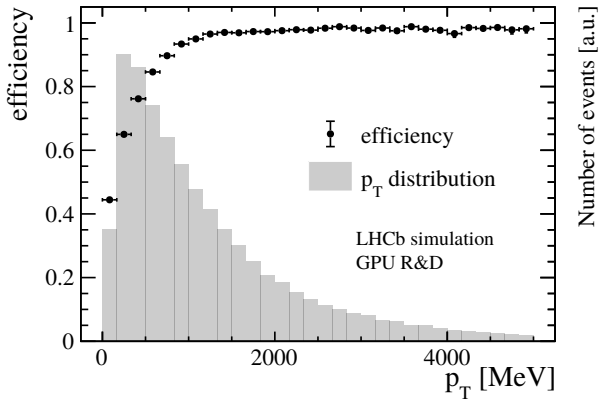


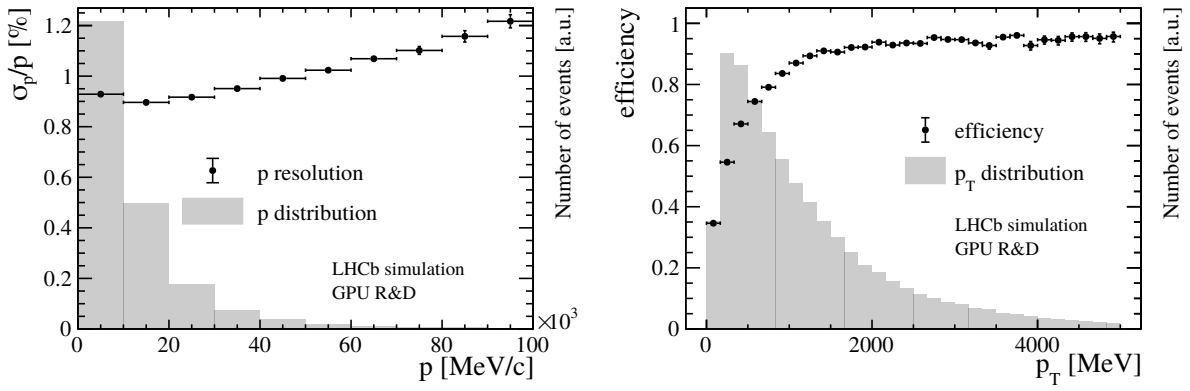
Figure 6: UT tracking efficiency as a function of p_{T}. The generated p_{T} distribution is shown in the shaded region. From Ref. [3].

3.3 UT Tracking

The Upstream Tracker (UT) is a silicon strip detector located upstream of the LHCb dipole magnet and consisting of four detector layers. The strips in the first and fourth layers are aligned parallel with the y-axis and are referred to as x layers. The second and third layers are rotated by $+5^\circ$ and -5°, respectively, and are referred to as u and v layers. The small magnetic field in the region allows the UT to provide an initial momentum estimate for charged tracks. VELO tracks are extrapolated to the UT and search windows are opened assuming $p > 3$ GeV, the minimum momentum required for muon identification with the LHCb detector. Track seeds are created using the first and third layers. If no seed is found, the fourth and second layers are used. Hits are added to the resulting seed to create three- or four-hit UT tracks. Figure 6 shows the UT tracking efficiency. The UT tracking algorithm is discussed in more detail in Ref. [7].

3.4 SciFi Tracking

The Scintillating Fiber (SciFi) tracker is located downstream of the LHCb dipole magnet and consists of twelve layers of scintillating fibers. The twelve layers are arranged into three tracking stations with four layers each. These layers have the same $x - u - v - x$ arrangement as the UT. Seed triplets are created



(a) (b)

Figure 7: The SciFi momentum resolution (a) and tracking efficiency (b). In both cases the generated distribution is shown in the shaded region. From Ref. [3].

using the x layers. Using these seeds, the hit positions on the remaining layers can be estimated to within a few millimeters. The forward tracking algorithm provides an improved momentum estimate. While forward tracking in HLT1 in Runs 1 and 2 required a minimum p_{T} of 500 MeV in order to limit the search area for forward track hits, Allen is able to reconstruct all tracks with $p > 3$ GeV with no p_{T} requirement. Figure 7 shows the SciFi tracking efficiency and momentum resolution.

3.5 Muon Identification

SciFi tracks are matched to hits in the muon stations in order to identify charged particles as muons. Allen uses the same algorithm used in Runs 1 and 2 at LHCb. This is documented in Ref. [8]. The muon identification efficiency is shown in Figure 8.

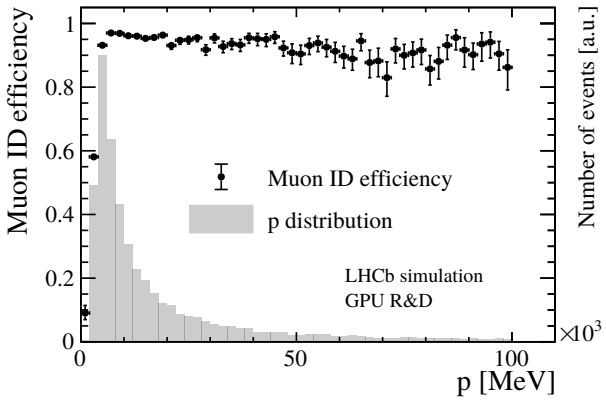


Figure 8: Muon identification efficiency for reconstructed SciFi tracks. From Ref. [3]

3.6 Kalman Filter

Allen uses a fast VELO-only Kalman filter to fit the reconstructed tracks. This fit uses a momentum-dependent parameterization to describe the noise from multiple scattering within the VELO, improving the track description at its point closest to the beamline. This provides an improved impact parameter

resolution and allows for better discrimination between prompt and displaced tracks. Figure 9 compares the performance of selections using this parameterized Kalman filter and a simplified Kalman filter using no momentum information. Fitting only the VELO segment of the track results in a fit that takes less than one percent of the total Allen execution time.

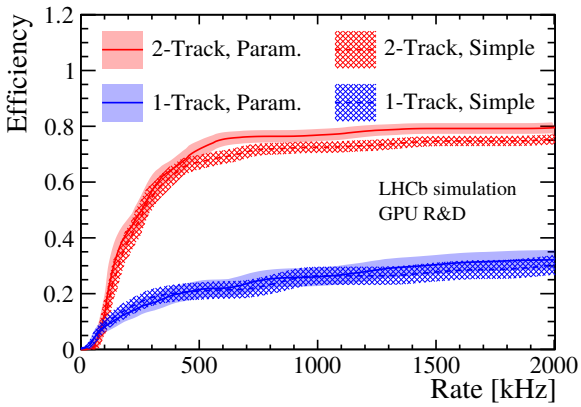


Figure 9: Selection efficiency for $B_s^0 \to \phi\phi$ decays using a Kalman filter with a momentum-dependent noise parameterization and a simple Kalman filter with no momentum dependence. The rates of the hypothetical selections are varied by adjusting the impact parameter requirement. From Ref. [3].

3.7 Selections

Selections in HLT1 are performed on one- and two-track candidates. Allen includes prototype HLT1 selections that cover most LHCb physics. While only five selections are shown here, Allen can perform (100) selections with minimal impact on throughput. These selections are tuned to accept an output rate of 1 MHz. Table 1a lists the output rates of individual selections. Table 1b shows selection efficiencies for various signal channels. These prototype selections are based on simple rectangular cuts. Experience from Runs 1 and 2 has shown that selection based on machine learning algorithms can provide improved efficiencies at the same rates. These prototype selections, however, already demonstrate the power of LHCb's software-only trigger strategy in Run 3. Allen provides trigger efficiencies of 30% to 50% for heavy flavor decays to fully hadronic final states, compared to the 5% to 10% percent shown in Figure 2.

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

(a)

Signal	Efficiency
$B^0 \to K^{*0}\mu^+\mu^-$	79 ± 3
$B^0 \to K^{*0}e^+e^-$	52 ± 4
$B_s^0 \to \phi\phi$	57 ± 3
$D_s^+ \to K^+K^-\pi^+$	35 ± 4
$Z \to \mu^+\mu^-$	77 ± 1

(b)

Table 1: Trigger rates (a) and efficiencies (b) for the prototype selections implemented in Allen. The trigger candidate causing the positive decision must be matched to a generated signal decay product to contribute to the efficiency. The efficiencies include losses due to a Global Event Cut that removes the highest occupancy events in order to decrease processing time.

4 Throughput

Allen will run on GPUs in LHCb's roughly 170 Event Builder nodes. Each node can host three GPUs, so Allen must be able to handle a 30 MHz event rate using fewer than 510 GPUs. Figure 10 shows Allen's throughput on several GPUs. This demonstrates that Allen can process events at just over 70 kHz on a GeForce RTX 2080 Ti GPU, allowing Allen to process the full LHC event rate using about 430 cards. Figure 10 also shows that Allen's performance scales well with theoretical GPU performance and improves with each new generation of GPUs. This indicates that Allen's performance will continue to improve before the beginning of Run 3.

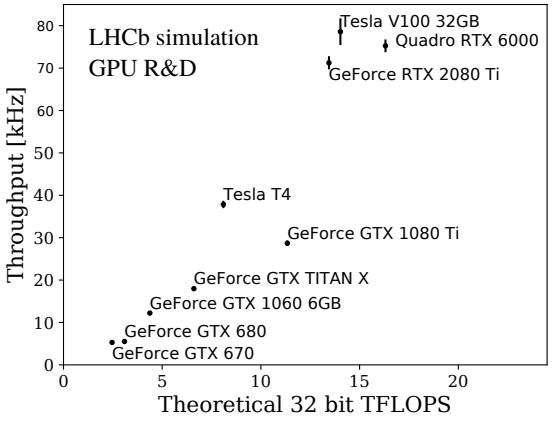


Figure 10: Throughput of the full Allen sequence for various GPUs as a function of each GPU's theoretical maximum 32-bit TFLOPS. From Ref. [3].

5 Future Prospects

5.1 Multi-track Vertices

Allen can reconstruct forward tracks with no p_{T} requirement. This could allow Allen to efficiently reconstruct three- and four-track vertices. Experience with the LHCb HLT2 topological trigger [9, 10] in Runs 1 and 2 has shown that selections based on multi-track vertices become more powerful than those based on one- and two-track candidates as tracking momentum requirements decrease. This could allow HLT1 to trigger with similar efficiencies at a lower trigger rate.

5.2 Calorimeter Reconstruction in HLT1

LHCb's HLT1 has only used information from the tracking systems in the past. Experience with VELO clustering algorithms in Allen indicates that calorimeter clustering could be efficiently implemented on GPUs. Implementing calorimeter clustering in Allen would allow for electron identification in HLT1. This would facilitate important measurements using electrons, such as tests of lepton universality and searches for dark photons decaying to electrons.

6 Conclusions

Allen is the first full software trigger stage implemented on GPUs for a high energy physics experiment. LHCb's baseline HLT1 has been fully implemented in Allen, and optimizations and improvements continue.

Furthermore, Allen will allow LHCb to increase the scope of its physics program in Run 3. As Allen's throughput continues to improve, it will be able to handle additional tasks. As a result, incremental improvements to throughput could lead to overhauled trigger strategies. These possibilities will continue to expand as GPUs improve before the beginning of Run 3.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. National Science Foundation Graduate Research Fellowship under Grant No. 1122374. This work is also supported by the U.S. National Science Foundation under Grant No. PHY-1912836.

References

[1] J. Alwall, et al., JHEP **07**, 079 (2014) doi:10.1007/JHEP07(2014)079 [arXiv:1405.0301 [hep-ph]].

[2] C. Fitzpatric and V. V. Gligorov, Tech. Rep. LHCb-PUB-2014-027. CERN-LHCb-PUB-2014-027, CERN, Geneva, 2014.

[3] R. Aaij, et al., Comput. Softw. Big Sci. **4**, no.1, 7 (2020) doi:10.1007/s41781-020-00039-7 [arXiv:1912.09161 [physics.ins-det]].

[4] LHCb Collaboration, Tech Rep. CERN-LHCC-2014-016, CERN, Geneva, 2018.

[5] LHCb Collaboration, Tech. Rep. CERN-LHCC-2020-006, CERN, Geneva, 2020.

[6] D. H. Cámpora Pérez, N. Neufeld, and A. Riscos Núñez, 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (2019) 698-707, doi:10.1109/IPDPSW.2019.00118.

[7] P. Fernandez Declara, et al., IEEE Access **7**, 91612-91626 (2019) doi:10.1109/ACCESS.2019.2927261 [arXiv:2002.11529 [physics.ins-det]].

[8] F. Archilli, et al., JINST **8**, P10020 (2013) doi:10.1088/1748-0221/8/10/P10020 [arXiv:1306.0249 [physics.ins-det]].

[9] V. Gligorov and M. Williams, JINST **8**, P02013 (2013) doi:10.1088/1748-0221/8/02/P02013 [arXiv:1210.6861 [physics.ins-det]].

[10] T. Likhomanenko, et al., J. Phys. Conf. Ser. **664**, no.8, 082025 (2015) doi:10.1088/1742-6596/664/8/082025 [arXiv:1510.00572 [physics.ins-det]].