A Comparison of CPU and GPU Implementations for the LHCb Experiment Run 3 Trigger

Citation: Computing and Software for Big Science. 2021 Dec 22;6(1):1

As Published: https://doi.org/10.1007/s41781-021-00070-2

Publisher: Springer International Publishing

Persistent URL: https://hdl.handle.net/1721.1/138852

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context https://doi.org/10.1007/s41781-021-00070-2

ORIGINAL ARTICLE

A Comparison of CPU and GPU Implementations for the LHCb Experiment Run 3 Trigger

R. Aaij[21] · M. Adinolfi[34] · S. Aiola[17] · S. Akar[43] · J. Albrecht[10] · M. Alexander[39] · S. Amato[1] · Y. Amhis[8] · F. Archilli[12] · M. Bala[25] · G. Bassi[20,55] · L. Bian[45] · M. P. Blago[31] · T. Boettcher[42] · A. Boldyrev[48] · S. Borghi[40] · A. Brea Rodriguez[29] · L. Calefice[9,10] · M. Calvo Gomez[49] · D. H. Cámpora Pérez[31,47] · A. Cardini[19] · M. Cattaneo[31] · V. Chobanova[29] · G. Ciezarek[31] · X. Cid Vidal[29] · J. L. Cobbledick[40] · J. A. B. Coelho[8] · T. Colombo[31] · A. Contu[19] · B. Couturier[31] · D. C. Craik[42] · R. Currie[38] · P. d'Argent[31] · M. De Cian[32] · D. Derkach[48] · F. Dordei[19] · M. Dorigo[20,56] · L. Dufour[31] · P. Durante[31] · A. Dziurda[24] · A. Dzyuba[26] · S. Easo[37] · S. Esen[9] · P. Fernandez Declara[31] · S. Filippov[27] · C. Fitzpatrick[40] · M. Frank[31] · P. Gandini[17] · V. V. Gligorov[9] · E. Golobardes[49] · G. Graziani[14] · L. Grillo[40] · P. A. Günther[12] · S. Hansmann-Menzemer[12] · A. M. Hennequin[31] · L. Henry[17,30] · D. Hill[32] · S. E. Hollitt[10] · J. Hu[12] · W. Hulsbergen[21] · R. J. Hunter[36] · M. Hushchyn[48] · B. K. Jashal[30] · C. R. Jones[35] · S. Klaver[21] · K. Klimaszewski[25] · R. Kopecna[12] · W. Krzemien[25] · M. Kucharczyk[24] · R. Lane[34] · F. Lazzari[20,55] · R. Le Gac[7] · P. Li[12] · J. H. Lopes[1] · M. Lucio Martinez[21] · A. Lupato[40] · O. Lupton[36] · X. Lyu[4] · F. Machefert[8] · O. Madejczyk[23] · S. Malde[41] · J. F. Marchand[6] · S. Mariani[14,31,51] · C. Marin Benito[31] · D. Martinez Santos[29] · F. Martinez Vidal[30] · R. Matev[31] · M. Mazurek[31] · B. Mitreska[40] · D. S. Mitzel[31] · M. J. Morello[20,53] · H. Mu[2] · P. Muzzetto[19,31] · P. Naik[34] · M. Needham[38] · N. Neri[17,52] · N. Neufeld[31] · N. S. Nolte[10,31] · D. O'Hanlon[34] · A. Oyanguren[30] · M. Pepe Altarelli[31] · S. Petrucci[38] · M. Petruzzo[17] · L. Pica[20,55] · F. Pisani[31] · A. Piucci[12] · F. Polci[9] · A. Poluektov[7] · E. Polycarpo[1] · C. Prouve[29] · G. Punzi[20,54] · R. Quagliani[9] · R. I. Rabadan Trejo[7] · M. Ramos Pernas[36] · M. S. Rangel[1] · F. Ratnikov[41,48] · G. Raven[22] · F. Reiss[9] · V. Renaudin[41] · P. Robbe[8] · A. Ryzhikov[48] · M. Santimaria[15] · M. Saur[10] · M. Schiller[39] · R. Schwemmer[31] · B. Sciascia[15] · A. Solomin[34,50] · F. Suljik[41] · N. Skidmore[40] · M. D. Sokoloff[43] · P. Spradlin[39] · M. Stahl[43] · S. Stahl[31] · H. Stevens[10] · L. Sun[45] · A. Szabelski[25] · T. Szumlak[23] · M. Szymanski[31] · D. Y. Tou[9] · G. Tuci[20,54] · A. Usachov[21] · N. Valls Canudas[28] · R. Vazquez Gomez[29] · S. Vecchi[13] · M. Vesterinen[36] · X. Vilasis-Cardona[49] · D. Vom Bruch[7] · Z. Wang[33] · T. Wojton[24] · M. Whitehead[34] · M. Williams[42,44] · M. Witek[24] · Y. Xie[5] · A. Xu[3] · H. Yin[5] · M. Zdybal[24] · O. Zenaiev[31] · D. Zhang[5] · L. Zhang[2] · X. Zhu[2] · The LHCb Collaboration[11,16,18,46]

Received: 18 May 2021 / Accepted: 20 September 2021

Abstract

The Large Hadron Collider beauty (LHCb) experiment at CERN is undergoing an upgrade in preparation for the Run 3 data collection period at the Large Hadron Collider (LHC). As part of this upgrade, the trigger is moving to a full software implementation operating at the LHC bunch crossing rate. We present an evaluation of a CPU-based and a GPU-based implementation of the first stage of the high-level trigger. After a detailed comparison, both options are found to be viable. This document summarizes the performance and implementation details of these options, the outcome of which has led to the choice of the GPU-based implementation as the baseline.

Keywords Real-time · Heterogeneous · High-throughput · Parallel computing · High-level trigger · Software

Introduction

The Large Hadron Collider beauty (LHCb) experiment is a general-purpose spectrometer instrumented in the forward direction based at the Large Hadron Collider (LHC) [1]. Although optimized for the study of hadrons containing beauty and charm quarks, LHCb's physics programme

To be submitted to Computing Software for Big Science.

L. Bian, L. Sun: Associated to Center for High Energy Physics, Tsinghua University, Beijing, China. A. Boldyre, D. Derkach, M. Hushchyn, F. Ratnikov, A. Ryzhikov: associated to Yandex School of Data Analysis, Moscow, Russia.

Extended author information available on the last page of the article

Published online: 22 December 2021

gradually expanded over the course of its first data collection period,[1] taking in Kaon physics, prompt spectroscopy, electroweak physics, and searches for putative heavy and long-lived exotic particles beyond the Standard Model.

The LHC provides a non-empty bunch crossing rate of up to 30 MHz, with a variable number of inelastic proton-proton interactions per bunch crossing (μ) which can be adjusted to suit the physics programme of each experiment. During Runs 1 and 2, LHCb took data with a μ of between 1.1 and 2.5, corresponding to an instantaneous luminosity of around $4 \times 10^{32}/\mathrm{cm}^2/\mathrm{s}$. A fixed-latency hardware trigger (L0), based on calorimeter and muon system information, reduced the 30 MHz LHCb bunch crossing rate to ~1 MHz at which the detector readout operated. These events were then passed to a two-stage asynchronous high-level trigger (HLT) system entirely implemented in software. In the first stage, HLT1, a partial event reconstruction selected events based on inclusive signatures, reducing the event rate by an order of magnitude. Accepted events were stored using a 11 PB disk buffer in order to align and calibrate the detector. After this procedure, events were passed to the second stage, HLT2, which had sufficient computing resources to run the full offline-quality detector reconstruction. A multitude of dedicated selections deployed in HLT2 reduced the data to an output rate of 12.5 kHz using a combination of full and reduced event formats [2].

The ambitious goal of the upgraded LHCb experiment in Run 3 (i.e. starting from 2022) is to remove the hardware trigger and directly process the full 30 MHz of data at an increased luminosity of $2 \times 10^{33}/\mathrm{cm}^2/\mathrm{s}$ in the HLT. At this luminosity, corresponding to a μ of around 6, it is no longer possible to efficiently identify bunch crossings of interest based purely on calorimeter and muon system information, as there is too much quantum chromodynamics (QCD) background generated by the pileup *pp* collisions [3]. It is necessary to instead fully read the detector out for every bunch crossing and fully perform the real-time processing in the HLT. This allows the much more sophisticated selections, in particular selections based on charged particle trajectories reconstructed in the whole of LHCb's tracker system, to be deployed already in HLT1. Such a full-software trigger will not only allow LHCb to maintain its Run 1 and 2 efficiencies for muonic signatures, but will lead to a factor two improvement in efficiency for hadronic signatures compared to the calorimeter-based L0 trigger, despite the harsher Run 3 environment.

LHCb's Run 3 data-gathering conditions imply a data volume of around 32 terabytes (Tb) per second, comparable to what the ATLAS [4] and CMS [5] software triggers will be required to process during high-luminosity LHC runs from 2027 onwards. The design and delivery of LHCb's high-level trigger is therefore also one of the biggest computing challenges that the field of high-energy physics is facing today. The closest current parallel to LHCb's system is that of the ALICE experiment [6], which will also operate a triggerless readout in Run 3 with an objective to reduce an input data rate of roughly 10 Tb/s to a manageable amount by performing a full detector reconstruction in real-time.

The concept of a pure CPU-based solution for this approach was reviewed during the preparation of LHCb's Trigger and Online TDR [7] in 2014 followed by a systematic rewrite of the LHCb trigger software infrastructure, which enabled data collection in these conditions. In parallel, R&D efforts have explored a possible usage of GPUs for HLT1 [8], referred to as the hybrid approach in the following. An intensive effort was launched to demonstrate if such a hybrid system could be delivered in time for Run 3 and concluded in a positive review of its TDR [9] in early 2020. After a detailed comparison of both options, the collaboration selected the hybrid approach as the new Run 3 baseline. This decision parallels that of ALICE, which pioneered the use of GPUs among LHC experiments during the last decade [10] and whose Run 3 triggerless readout and full real-time reconstruction mentioned earlier will be mainly implemented on GPUs.

This document compares both options and summarizes the salient points which led to the decision. It reflects the status of both implementations at the time the decision was made, in April 2020. Further significant improvements [11] in both throughput and physics performance, which will not be discussed in this document, have been achieved since then.

This document is structured as follows. In the "Introduction" section, the data acquisition (DAQ) and HLT architecture of both systems is summarized. The "DAQ and HLT Architectures" section describes the boundary conditions within which the implemented HLT1 triggers must operate, including available financial resources and operational constraints. The section "HLT1 Sequence" describes the HLT1 trigger sequence and algorithms, which are the basis of the performance comparison. The "Throughput" section summarizes the performance of the two architectures in terms of throughput, while the "Physics Performance" section presents their physics performance. The "Cost–Benefit Analysis" section combines the performance assessment into a cost–benefit analysis.

[1] The first data collection period was broken into two "runs", with Run 1 taking place from 2009 to 2013 and Run 2 taking place from 2015 to 2018.

DAQ and HLT Architectures

LHCb's DAQ and event building (EB) infrastructure is described in the Trigger and Online Upgrade Technical Design Report [7]. The full detector is read out for all LHC bunch crossings, and information from subdetectors is received by around 500 custom backend field-programmable gate array (FPGA) "TELL40" boards hosted in a farm of EB servers, with three TELL40 boards per server. These subdetector fragments are then assembled into "events", with one event corresponding to one LHC bunch crossing, and sent to the HLT for processing. Both the event building and HLT are fully asynchronous and no latency requirements exist in the system.

The following two HLT processing architectures are under consideration.

1. *CPU-only* which implements both HLT1 and HLT2 using Event Filter Farm (EFF) CPU servers.
2. *Hybrid* which implements HLT1 using GPU cards installed in the EB servers with the CPU-based HLT2 running as before on the EFF.

The HLT2 software and processing architecture are identical in both cases.

CPU-Only Architecture

The CPU-only architecture is illustrated in Fig. 1. Briefly, it consists of

- A set of custom FPGA cards, called TELL40, which together receive on average 32 Tb/s of data from LHCb's subdetectors;
- A set of EB servers which host the TELL40 cards and implement a network protocol to bring the subdetector data fragments produced in a single LHC bunch crossing ("event" in LHCb nomenclature) together and then group O(1000) of these events into multi-event packets (MEPs) to minimize I/O overheads further down the line. The EB servers will be equipped with 32-core AMD EPYC (7502) CPUs;
- An EFF illustrated on the bottom left of Fig. 1 which receives MEPs from the EB and executes HLT1. The memory available in the EB servers and HLT1 EFF nodes allows data to be buffered for $O(20)$ s in case of temporary network or processing issues. The HLT1 EFF servers are assumed to be equipped with the same 32-core AMD EPYC (7502) CPUs as the EB servers. When LHCb is not taking data this HLT1 EFF can also receive events from the disk servers (described below) and run the HLT2 process on them;
- A high performance network, including high-speed network interface cards (NICs) located in the EB servers and EFF nodes as well as a large switch, connects the EB and HLT1 EFF and allows transmission at a rate of 32 Tb/s.
- The HLT1 process reduces the event rate by a factor of between 30 and 60, and an array of disk servers buffers this HLT1 output data while the detector alignment and calibration are performed in quasi-real-time. This disk buffer, whose size is a tunable parameter of the system as discussed later in the "Cost–Benefit Analysis" section, allows HLT1 output data to be buffered for $O(100)$ h in case of problems with the alignment and calibration which require specialist intervention;
- A second EFF receives events from the disk servers once the alignment and calibration constants are available and runs the HLT2 process on them. Because of limitations in network bandwidth this second EFF cannot be used to process HLT1.

Advances in server technology have permitted a substantial reduction of the number of servers required by the EB, from 500 in the TDR [7] to around 173. This allows three TELL40 cards to be hosted per EB server rather than the one card foreseen in the TDR. This improvement means that the EB will be much more compact, and as a consequence, easier to upgrade in the future.

Hybrid Architecture

The hybrid architecture is illustrated in Fig. 2. It follows the same processing logic as the CPU-only solution: the full detector data is received by TELL40 boards and assembled into MEP packets by the EB servers, those MEP packets are then sent for processing by HLT1 and the events selected by HLT1 are recorded to the disk buffer for later processing by HLT2. Compared to the CPU-only solution it replaces the HLT1 EFF by GPU cards running HLT1, which are installed in the spare PCI express slots available in the EB servers. This allows HLT1 to reduce the data rate at the output of the EB by a factor of 30–60. This reduction in turn allows communication between the EB and EFF using a lower bandwidth (and consequently cheaper) network and removes the need to buy and install dedicated NICs in the EB and HLT1 EFF servers as they are already equipped with on-board 10 Gb interfaces. For the same reason, a much smaller switch is required to handle the data traffic between HLT1 and the disk servers. HLT2 then runs similarly to the CPU-only solution on the EFF.

Real-Time Analysis

In Run 2, LHCb successfully adopted a real-time analysis model, which is documented in detail in references [12, 13].

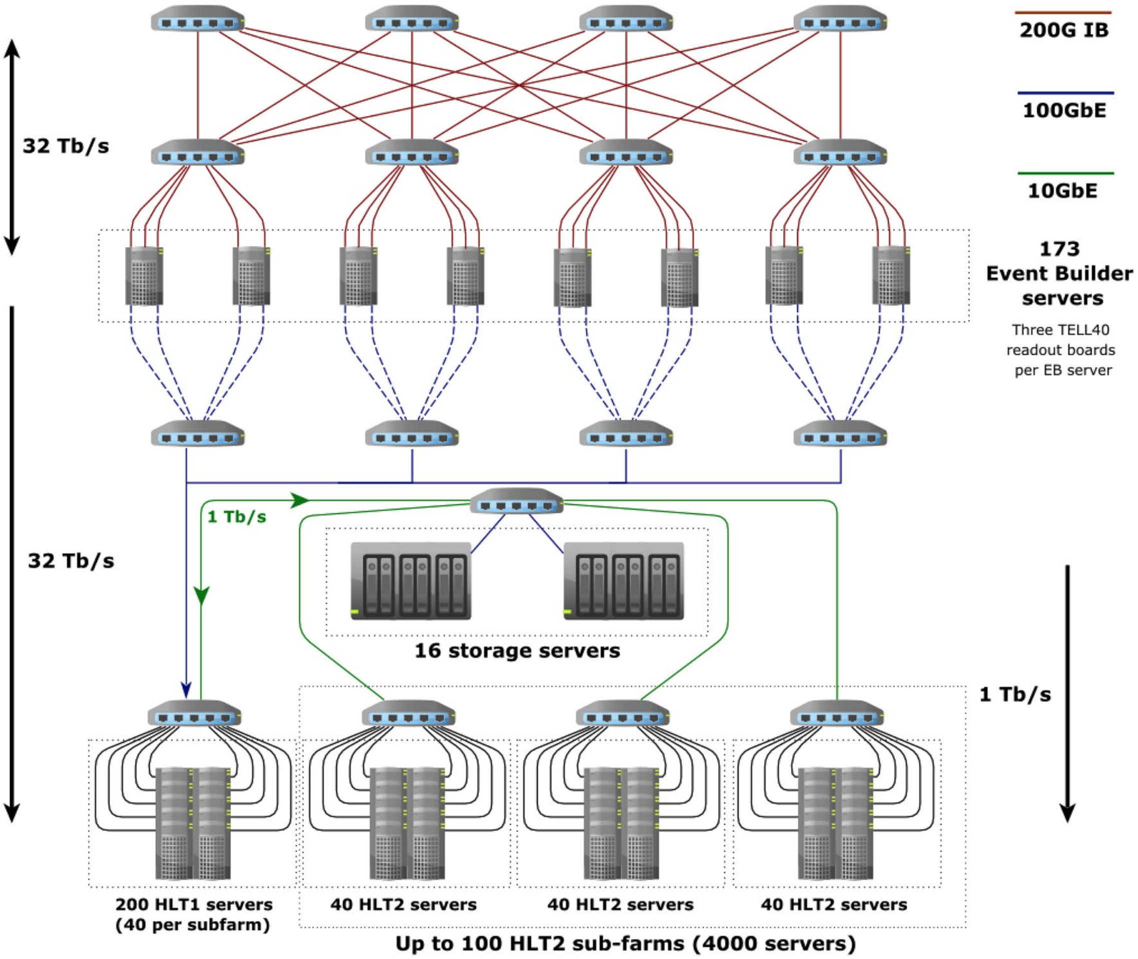

Fig. 1 CPU-only architecture of the Run 3 DAQ, including the Event Builder, the Event Filter Farm and dedicated storage servers for the disk buffer. The network between the storage servers and HLT1 servers (leftmost green line) is bidirectional, allowing the HLT1 servers to be used for HLT2 processing when there are no LHC collisions happening. The label "200G IB" refers to the Infiniband link between the detector and EB servers, while "100 GbE" and "10 GbE" refer to Ethernet links of 100 Gb/s and 10 Gb/s, respectively

The most important aspects relevant for the comparison presented in this document are summarized here:

- To make optimal use of limited offline resources, around three quarters of LHCb's physics programme is written to the TURBO stream [14], a reduced format, which on one hand allows flexible event information to be added in a selective manner and on the other hand can discard a user-specified fraction of both the raw and reconstructed detector data.
- Consequently, HLT2 must be able to run the complete offline-quality reconstruction. Therefore, HLT2 must use the highest quality alignment and calibration at all times. A substantial disk buffer must therefore be purchased to allow events selected by HLT1 to be temporarily saved while the full detector is aligned and calibrated in real-time. This buffer must be big enough not only to cover the steady-state data taking conditions but also to permit recovery from unforeseen operational issues in a reasonable timescale without loss of data.

Assumptions and Boundary Conditions

Having described the overall design of LHCb's Run 3 DAQ and HLT, as well as the processing technologies under consideration, we will now describe the boundary conditions which these technologies have to respect, as well as common assumptions relevant to the cost–benefit comparison in the "Cost–Benefit Analysis" section.

Use of Storage and Computing Resources During and Outside Data Collection

Throughout this document, it is assumed that the LHC is in "data-taking" mode for 50% of the year, and in either

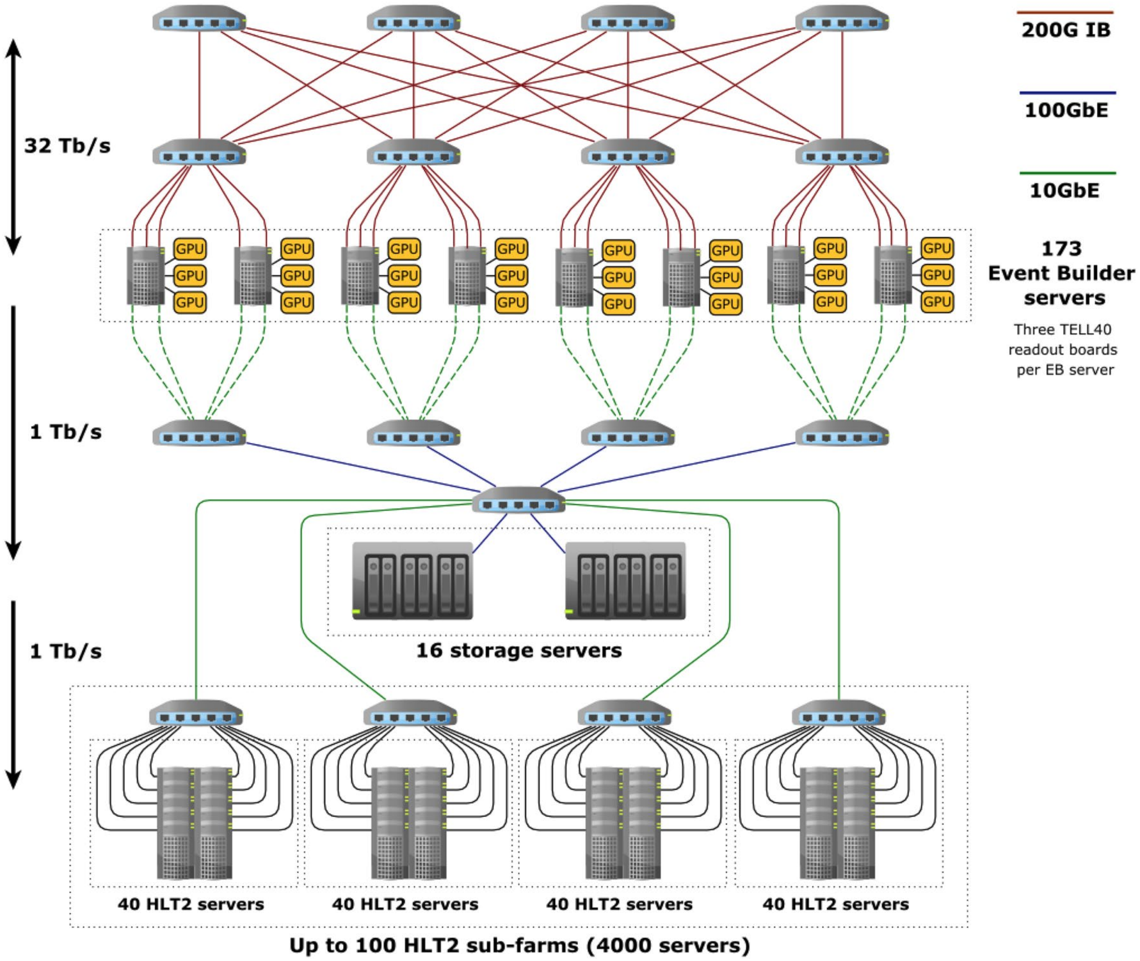

Fig. 2 Run 3 DAQ architecture in the case of the hybrid solution, with GPUs placed in the EB servers to reduce the data rate. Labels are the same as in Fig. 1

the winter shutdown or longer technical stops for the other 50%. When in data-taking, it is assumed that the LHC is in stable beams 50% of the time. During data-taking, it is assumed that all CPU resources are used to process HLT1 and/or HLT2. Outside data-taking, it is assumed that all HLT CPU resources are used to produce simulations for LHCb analyses. GPU resources can only be used to process HLT1 in-fill, and cannot be used opportunistically during data-taking. They cannot yet be used for producing simulations, and there is no realistic prospect of this changing on a short timescale. However, in principle, GPU resources could be used outside of data-taking if use cases can be found, as discussed in Ref. [9]. When LHCb is not taking data, the EB nodes will be used to produce simulations. However, they will not be used for any task other than event building while collecting data, as all the available memory bandwidth in these nodes will be required for the event building process and transferring data to the HLT1 application.

Existing and Pledged HLT2 Computing Resources

We quantify the computing resources available for HLT2 in terms of a reference QuantaPlex (“Quanta”) server consisting of two Intel E5-2630v4 10-core processors, which was the workhorse of our Run 2 HLT. This reference node corresponds to approximately 380 HEPSPEC.[2] We currently have roughly 1450 such equivalent servers available for Run 3 processing, with a further 1200 equivalent servers pledged, corresponding to a total capacity of slightly more than one million HEPSPEC. These servers can only be used

[2] The HEPSPEC benchmark is defined at http://w3.hepix.org/benchmarking. In reality, our HLT2 farm will consist of a mixture of servers of different generations and with different numbers of physical cores, but because of the asynchronous nature of HLT2 processing, load-balancing between these servers is an implementation detail and it is more convenient to quantify the available resources in units of the reference node as if the system were fully homogeneous.

to process HLT2 as it would not be cost-effective to equip so many old servers with the high-speed NICs required to process HLT1. So far no economical way has been found to reuse the predominantly very small disk drives in the old Run 2 servers, so there are no free storage resources available.

Event Building and Data Flow

In the CPU-only scenario no data reduction happens before the EFF, so all the data collected in the building network has to be distributed to the HLT1 CPU nodes at 32 Tb/s. Using newly available AMD 32-core CPUs a cost-effective implementation is to use a dual-socket server with a total of 64 physical cores and two network interfaces of 100 Gb/s. Each EB node requires two high-speed network connections for sending the event fragments between EB nodes while they are built. In addition, the distribution network needs to connect all HLT1 CPU nodes to the EB nodes as well as to at least 30 storage servers. These connections need to be optical.

In the hybrid scenario the GPUs hosted within the EB nodes execute the HLT1 process and reduce the data rate, so that only 0.5–1.0 Tb/s has to be sent from the EB to the disk buffer servers. The EB servers can therefore use their on-board 10 Gigabit interfaces to send the data and the distribution network needs significantly fewer optical connections.

Disk Buffer Boundary Conditions

The disk buffer needs to be able to handle at least 1 MHz of events coming from HLT1, with a potential upgrade to be able to handle 2 MHz as Run 3 progresses. A typical minimum bias event in LHCb Run 3 conditions is expected to be around 100 kB; however, events selected by HLT1 are bigger than average minimum bias events since they typically contain a hard QCD process leading to the production of a heavy flavour hadron. Therefore, assuming an event size of 120 kB to account for this effect, this implies 120 GB/s both for writing events coming from HLT1 and for reading events out into HLT2. The nominal rate of a 12 TB disk is 100 MB/s, thus 50 MB/s for writing and 50 MB/s for reading. However, the read and write speed of hard disks decreases as they fill up. As the system must be designed so that there is no dead time even when the buffer is almost full, an effective sustainable write and read rate of a single 12 TB disk is assumed to be 35–40 MB/s.

Since this part of the system is hardware-limited and must be able to handle burst data rates, a minimum of 2880 physical disks is required assuming minimal redundancy, and 3120 physical disks with adequate redundancy. A final point to note on the disk buffer is that the usable disk sizes are only around 80% of the nominal disk size, so the 12 TB disks actually only provide around 9.6 TB of usable storage each. In practice, the cost of such a minimal disk buffer is so large compared to the overall budget discussed earlier that spending money on buying additional disks or bigger disks is not really an interesting option.

HLT1 Sequence

The CPU-only and hybrid approaches under study in this document implement an HLT1 configuration which broadly corresponds to the one used in Run 2 [2] and whose physics objectives have been described in reference [9]. The reconstruction consists of the following components, whose performance in terms of efficiency, purity and resolution is discussed in the "Physics Performance" section.

- Vertex Locator (Velo) detector decoding, clustering, and reconstruction. Conceptually very similar algorithms are used here in the CPU and the GPU implementation. Only minor differences in physics performance are expected due to a limited number of architecture-specific optimizations.
- The primary vertex (PV) finding with tracks reconstructed in the Velo. Again only minor differences are expected in the physics performance of the CPU- and GPU-based implementations.
- Decoding of the UT and SciFi tracking detector raw banks. The raw bank formats are converted to the input of the pattern recognition algorithms. These are specific to the data layout and event model used by each architecture.
- Reconstruction of Velo-UT track segments. Different algorithms are used here for the CPU and the GPU implementations, which lead to slight differences in the physics performance.
- Reconstruction of long tracks[3] starting from reconstructed Velo-UT track segments. Both the CPU and GPU tracking algorithms use a parameterization of particle trajectories in the LHCb magnetic field and the initial Velo-UT momentum estimate[4] to speed up their reconstruction. One major difference in strategy is that the CPU algorithm matches Velo-UT track segments to doublets in the SciFi x-layers, while the GPU algorithm matches them to triplets. In addition, the CPU algorithm

[3] Long tracks are tracks that traverse the entire tracking system from Velo to SciFi. They deliver the best parameter estimate in terms of position and momentum and thus are the most valuable tracks for physics analysis.

[4] The momentum resolution of Velo-UT tracks is about 15% with significant non-Gaussian tails due to the small and inhomogeneous B field between Velo and UT.

applies a 490 MeV transverse momentum threshold when defining its search windows. These choices, and the differences in the Velo-UT algorithms, lead to somewhat different performance.

- Decoding of the muon raw banks and calculation of crossing points in the muon system, as well as implementation of the muon identification algorithm. Both architectures use very similar algorithms here.
- A simplified Velo-only Kalman filter, which uses the momentum estimate from the forward tracking to calculate a covariance matrix and estimate an uncertainty on the track impact parameter. Again, the underlying algorithm is the same for both implementations.

The HLT1 sequence described covers most use cases required by bottom and charm physics. There are, however, a few dedicated algorithms missing, such as a reconstruction for high momentum muons for electroweak physics, reconstruction of beam gas events or a reconstruction of low momentum tracks down to a transverse momentum of 80 MeV, which is motivated by the strange physics programme. While preliminary versions of these algorithms were ready in time for this comparison, they were not yet fully optimized in the same way as the other described algorithms. We did use these preliminary versions to test whether their inclusion impacted on the relative throughput between the CPU and GPU implementations, and found that the relative slowdown from including these algorithms was similar across the two architectures. It is therefore expected that these missing components will not change the conclusions of this document.

In addition to the reconstruction algorithms listed above, both the CPU and GPU HLT1 implement a representative subset of HLT selections including the finding of displaced vertices, monitoring, as well as writing of decision and selection reports. While the specific components are not exactly the same in both cases, these parts of the system consume comparatively little throughput and hence the differences are not relevant for this comparison.

A global event cut that removes 7% of the busiest minimum bias events is applied in all selections before the reconstruction is performed for both the CPU and GPU HLT1 implementations. The criterion used is that the total number of hits in the UT and SciFi tracking detectors be below a certain value. This criterion is chosen for historical reasons and because the UT and SciFi reconstructions are the most sensitive to occupancy, especially when trying to reconstruct low p_T signatures. However, in practice, any subdetector occupancy could be used, and LHCb will likely use the occupancy of whichever subdetector shows the best data-simulation agreement once the new detector is commissioned in Run 3. The impact of this criterion on the physics is given in the "Global Event Cut Efficiency" section.

Throughput

We define throughput as the number of events which can be processed by a given architecture per second in steady-state conditions, that is to say, neglecting the time it takes to initialize the HLT1 application at the start of each data collection period. They can be converted into GB/s by multiplying by the average expected Run 3 event size of 100 kB. An event is processed when it is read into HLT1, the HLT1 reconstruction and selection sequences used to decide whether or not to keep this event, and the event is finally written out (or not). The throughput of the CPU and GPU HLT1 implementations are measured using the same minimum bias samples. The GPU throughput is measured directly on the candidate production card. The CPU throughput is measured on both the Quanta reference nodes used for HLT2, and the dual-socket AMD 7502 EPYC nodes used for HLT1. While the HLT1 throughput measurements include both reconstruction, selection, and saving of trigger candidates, the HLT2 throughput measurement only includes the reconstruction and not the selection and saving of trigger candidates. Additional costs associated with these missing components of HLT2 are expected, but we expect that they can be counterbalanced with future performance improvements in the reconstruction.

The measured throughputs used in the rest of this document are

- CPU HLT1 : 171 kHz;
- GPU HLT1 : 92 kHz;
- CPU HLT2 on an AMD EPYC node : 471 Hz;
- CPU HLT2 on a Quanta node : 134 Hz.

Although we will discuss the cost–benefit of the two architectures later in the "Cost–Benefit Analysis" section, we can already conclude that both the CPU and GPU HLT1 architectures can be implemented using rather compact systems of $O(170)$ CPU servers or $O(330)$ GPU cards.

Physics Performance

This section presents key figures which are evaluated in a like-for-like comparison of the GPU and CPU performance. This includes performance numbers for track reconstruction and fitting, for PV reconstruction, for Muon-ID and the overall HLT performance for some representative trigger selections. Identical algorithms are used to fill the histograms and produce the plots based on the output of HLT1. The source code is compiled to operate on both the GPU and the CPU. The output is also translated to the same format. This ensures that the same definitions of physics performance

Table 1 Simulated samples used to evaluate the physics performance

Sample	No. events
$J/\psi \to \mu^+\mu^-$	10 k
$B_s^0 \to \phi\phi$	10 k
$B^0 \to K^{*0}e^+e^-$	10 k
$B^0 \to K^{*0}\mu^+\mu^-$	10 k
$D_s \to K\pi\pi$	10 k
$Z \to \mu^+\mu^-$	10 k
Minimum bias	10 k

All samples are simulated using the same pileup assumed in the Run 3 minimum bias samples

parameters such as efficiencies and resolutions are used when doing the comparison. The CPU compiled version of the GPU code has been checked to give results which agree to within 10^{-4}–10^{-3} with results obtained with the GPU version. More details of this comparison can be found in reference [9].

An overview of the samples used for these studies is given in Table 1. The specific samples used for the individual studies are listed in the corresponding subsections.

Tracking Efficiencies and Ghost Rates

All plots shown in this section are based on the B and D signal samples in Table 1. Tracking efficiencies are defined as the number of reconstructed tracks out of the number of reconstructible tracks in a given subsample. In this section, only particles which are reconstructible as long tracks are studied, which essentially means that they have left signals on at least 3 pixel sensors in the Velo and one x and one stereo cluster in each of the 3 fibre tracker (FT) stations. Furthermore, the studies are restricted to B and D daughter particles which are in the range $2 < \eta < 5$, which have true transverse momenta $p_\mathrm{T} \geq 0.5$ GeV/c and true momenta $p \geq 3$ GeV/c. Electrons are explicitly excluded because their performance has not yet been optimized to the same extent as that of hadrons and muons; however, we have checked the electron performance for both the CPU and GPU implementations of HLT1, and it does not change our conclusions.

Efficiencies to reconstruct these particles in the Velo, in both the Velo and UT, and in the Velo, UT and FT are shown as functions of the true p and p_T of the particles in Fig. 3. The performance of both technologies is very similar; the only difference is seen in the momentum dependence of the long track reconstruction efficiency. This difference is minor in the context of the overall trigger efficiency, since the majority of HLT1 triggers use only a subset of tracks coming from a given signal decay to select an event, and are therefore inherently robust to differences in single-track reconstruction efficiencies at the level of a few percent. While the CPU implementation is more efficient at low momenta, the GPU implementation is better at high momenta, and the overall single-track efficiency integrated over the kinematic range of interest agrees to better than 1% between the two algorithms. This difference is expected to be entirely related to the tuning of the algorithms and not to the underlying technology.

To check for a possible decay time bias in the Velo reconstruction, the Velo efficiency for long reconstructible tracks is studied as a function of the distance of closest approach to the beamline, *docaz* and as a function of the z position of the primary vertex in the event (Fig. 4). Again, the performance of both implementations is very similar. The loss in efficiency at large *docaz* is caused by the use of search windows which favour tracks coming from the beamline, and can be recovered at a moderate throughput cost.

Finally, Fig. 5 shows the fraction of ghost tracks among all long tracks. Ghost tracks are tracks which cannot be assigned to a true particle, thus are fake combinations of signals in the detector. Again, the performance of both implementations is very similar.

Track Parameter Resolution

For the comparison of the impact parameter resolution, $\sigma_{\mathrm{IP}x}$, and the momentum resolution, σ_p/p, the minimum bias sample (Table 1) is used. The results are shown in Fig. 6. Note that the x and y components of the impact parameter have very similar resolution; therefore, only $\sigma_{\mathrm{IP}x}$ is shown here. The impact parameter resolution is very similar for both technologies. The momentum resolution is worse in the GPU framework, with a maximum absolute resolution difference of 0.15–0.2% at low momenta. This difference is caused by a suboptimal tuning of the parameterization used to derive the momenta of the particles in the GPU algorithm. However, since the computational cost of this parametrization is negligible compared to the track finding itself, this difference in performance can be recovered without any significant change in its throughput.

Primary Vertex Reconstruction Efficiency and Resolution

A simulated primary vertex is defined as reconstructed if a primary vertex is found within 2 mm of its true position. The primary vertex efficiency as a function of the number of its reconstructed Velo tracks and of the primary vertex z position is shown in Fig. 7. All plots in this section are obtained on a sample of minimum bias events (Table 1).

The primary vertex resolution in x, y and z are studied as a function of the number of Velo reconstructible particles associated to the primary vertex and as a function of the z position of the primary vertex. The results on minimum bias

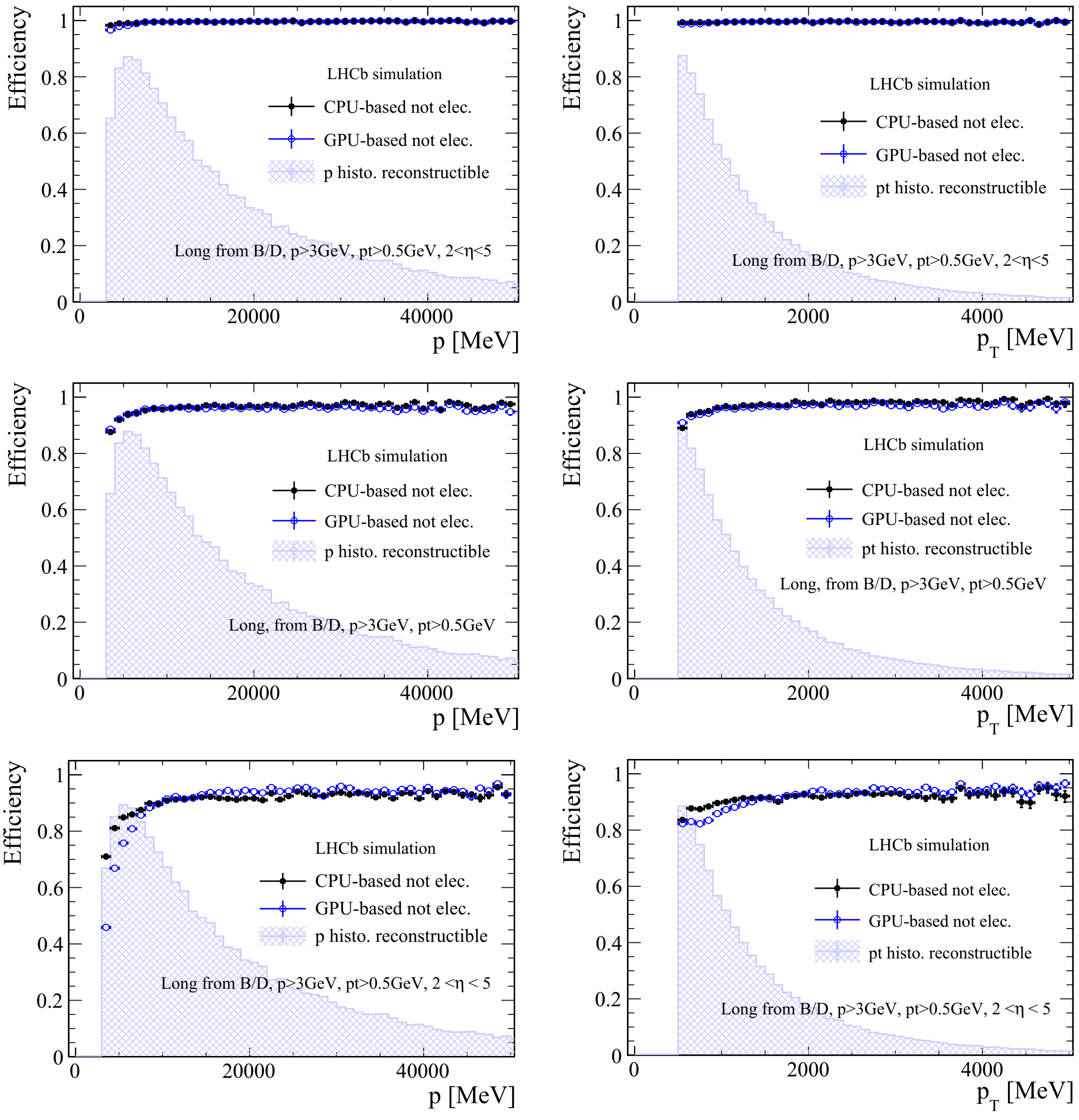

Fig. 3 Track reconstruction efficiency versus momentum, p (left) and transverse momentum, p_T (right) for long reconstructible particles from B and D decays within $2 < \eta < 5$. The reconstruction efficiency in the Velo (top), as Velo-UT track (middle) or as long track (bottom) are shown

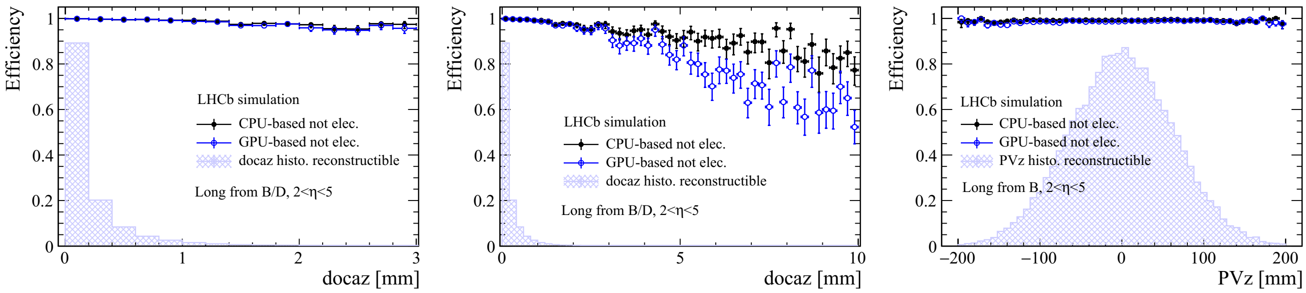

Fig. 4 Reconstruction efficiency for long reconstructible particles from B and D decays within $2 < \eta < 5$ in the Velo as a function of distance of closest approach to the beam axis (*docaz*) and the z position of the primary vertex (PVz). The *docaz* and z distribution of the long reconstructible particles are shown as well

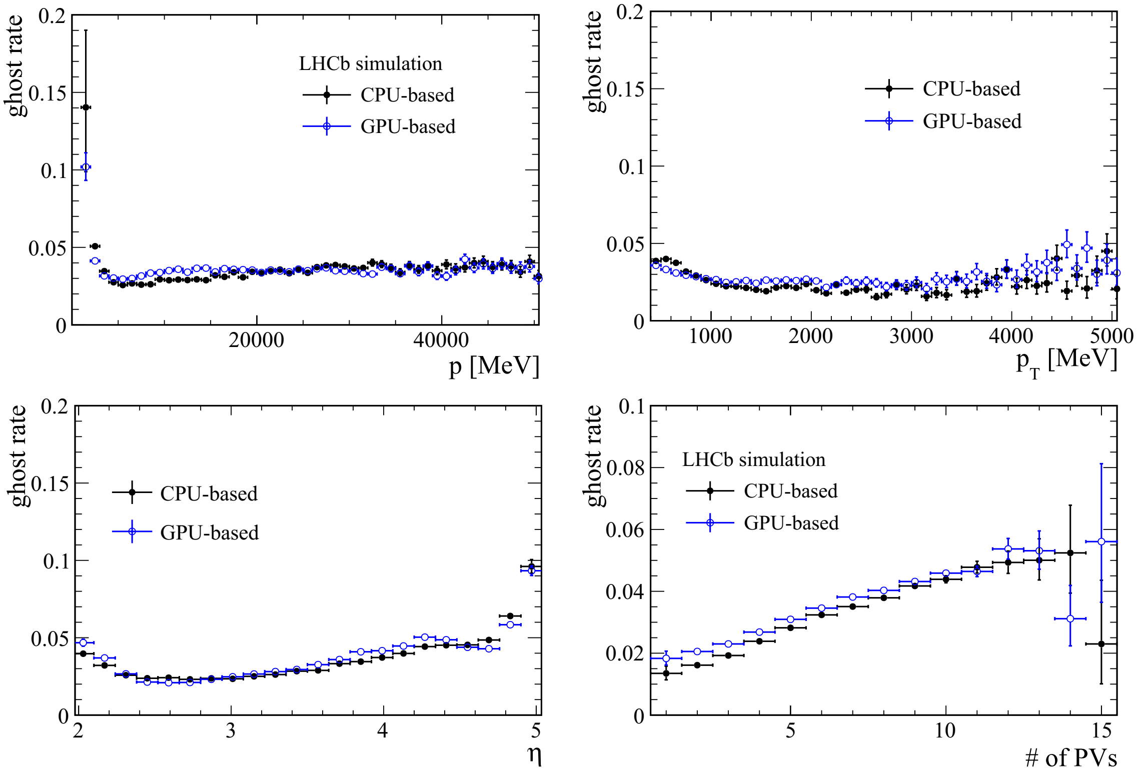

Fig. 5 Ghost rate of long tracks as a function of momenta, p, transverse momenta, p_{T}, and pseudo-rapidity, η, of the reconstructed tracks and the number of primary vertices, *# of PVs*, in the event

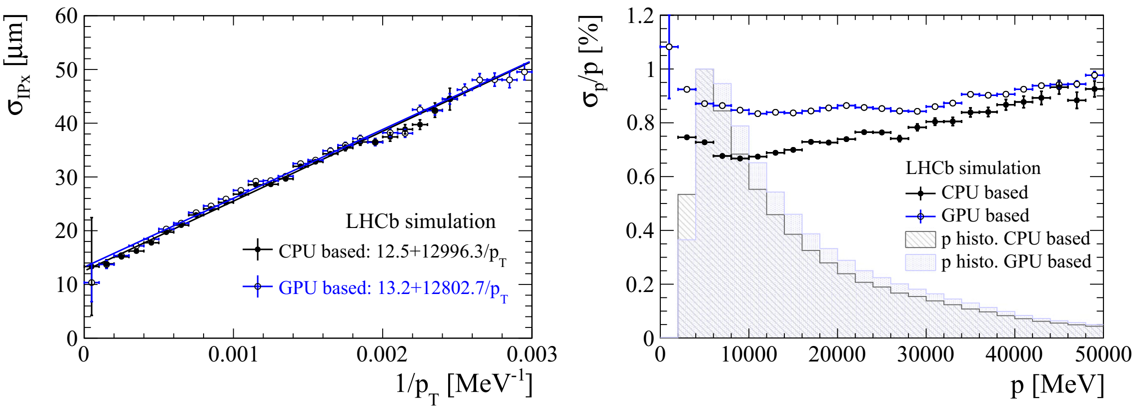

Fig. 6 Resolution of the x projection of the impact parameter, $\sigma_{\mathrm{IP}x}$ as a function of the inverse transverse momentum, $1/p_{\mathrm{T}}$, and the relative momentum resolution σ_{p}/p as a function of the momentum, p

events are shown in Fig. 8. The x resolution of the primary vertex is very similar to the y resolution; thus only one of them is shown. The average resolution of the minimum bias data set is $\sigma_x = \sigma_y \sim 14$ μm and $\sigma_z \sim 87$ μm for both technologies. The performance in terms of both efficiency and resolution is close to identical for the two implementations.

Muon ID Efficiency

The efficiency for the muon identification has been measured using the $J/\Psi \to \mu^+\mu^-$, $Z \to \mu^+\mu^-$ and $B^0 \to K^{*0}\mu^+\mu^-$ samples (Table 1). The denominator of the efficiency term counts muons with a minimum momentum of $p \geq 3$ GeV/c and a minimum transverse momentum of $p_{\mathrm{T}} \geq 0.5$ GeV/c in

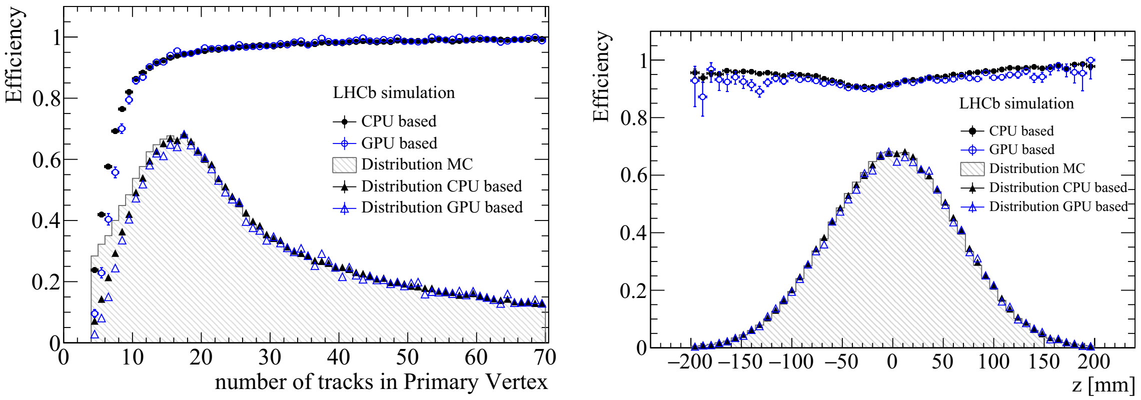

Fig. 7 Efficiency to reconstruct primary vertices as function of the number of reconstructed Velo tracks associated to the simulated primary vertex and as function of the true vertex *z* position

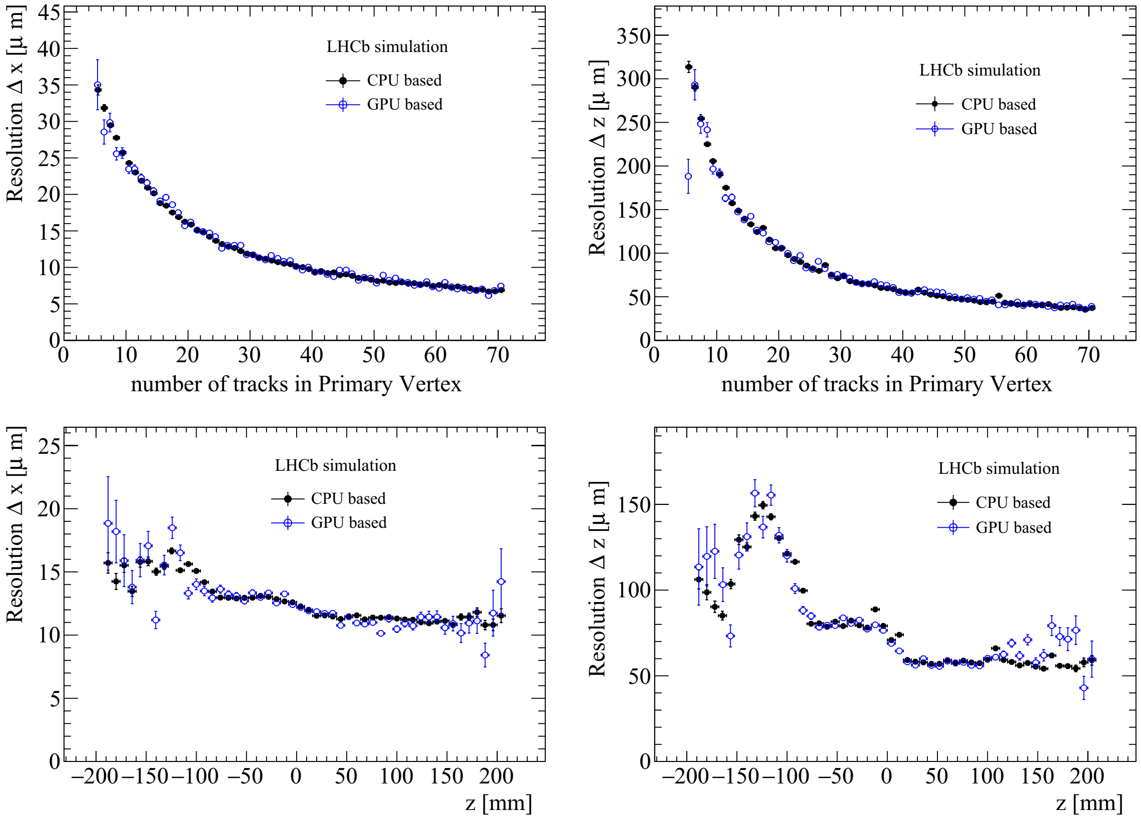

Fig. 8 Resolution in *x* and *z* of all reconstructed primary vertices as function of the number of reconstructed Velo tracks associated to the simulated primary vertex (top row) and as function of the true vertex *z* position (bottom row)

the pseudorapidity range $2 \leq \eta \leq 5$. Furthermore, they must be reconstructed as long tracks. The numerator additionally requires these tracks to be identified as a muon in the detector. The efficiency is displayed in Fig. 9 as a function of the momentum of the muon and as function of its pseudorapidity. A second performance criterion is the purity of the muon reconstruction. For this we count which fraction of pions with a minimum momentum $p \geq 3$ GeV/c and a minimum transverse momentum of $p_{\mathrm{T}} \geq 0.5$ GeV/c in the pseudorapidity range $2 \leq \eta \leq 5$ are misidentified as muons by the reconstruction algorithm (Fig. 10). The GPU implementation of the muon identification gives better performance than the CPU version, with an absolute efficiency improvement of up to 10% at small pseudorapidities. The misidentification

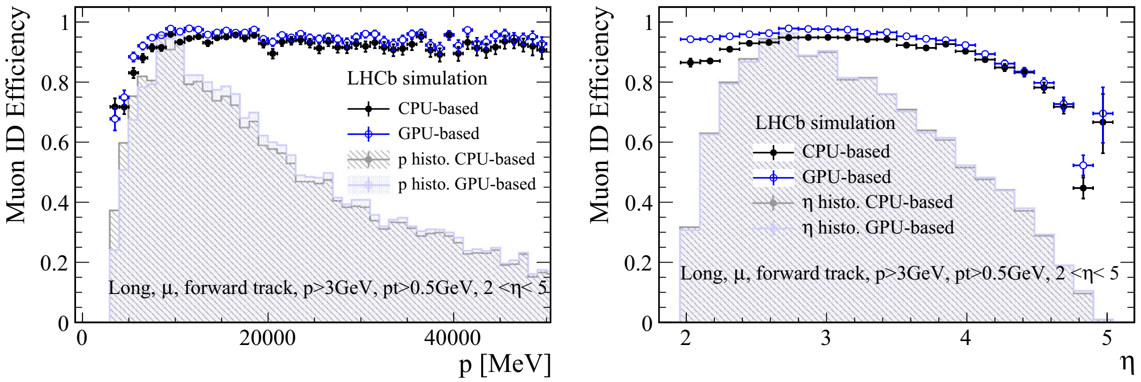

Fig. 9 Efficiency to identify true muons reconstructed as long muon tracks as muons as function of the muon momentum and pseudorapidity

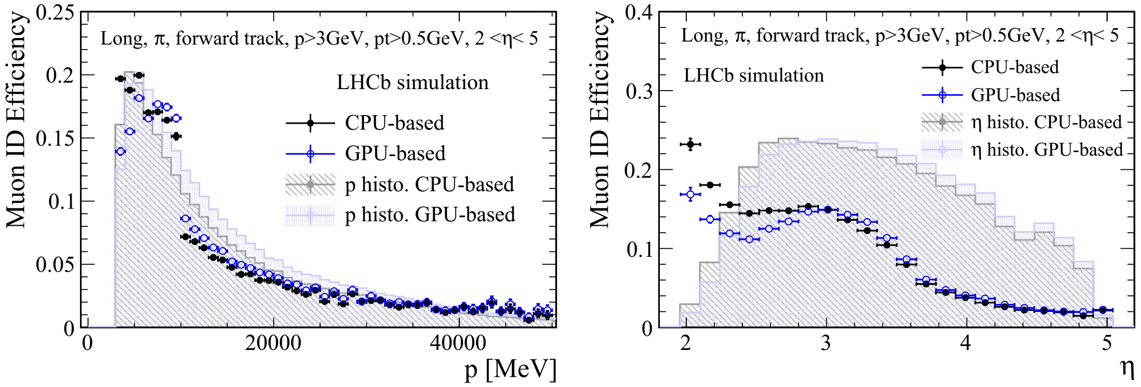

Fig. 10 Efficiency to falsely identify true pions reconstructed as long muon tracks as muons as function of the pion momentum and pseudorapidity

rate is similarly better by up to 5% absolute at small pseudorapidities and momenta for the GPU implementation, while it is a couple of percent better at higher momenta for the CPU implementation. It is expected that it is a matter of throughput-neutral tuning to obtain similar results for the two implementations.

HLT1 Efficiency

In this part of the study, the full HLT1 sequence of both the CPU and the GPU implementations are run on the simulated signal samples listed in Table 1, and the trigger efficiencies are then compared. The efficiency denominator contains all signal candidates for which all stable charged daughter particles are reconstructible as long tracks within the detector acceptance of $2 < \eta < 5$. Thus, the trigger efficiency is the number of selected events divided by the total number of events. We choose four generic trigger selections to perform the comparison: a one-track MVA line (`TrackMVA`), a two-track MVA line (`TwoTrackMVA`), a low-mass dimuon line (`DiMuonLowMass`) and a single, high-p_{T} muon line (`SingleHighPtMuon`). The former two are inclusive selections designed to select generic B and D decays, the third is specialized for semileptonic B decays, whereas the latter is more suitable for high-p_{T} electroweak physics such as $Z \to \mu^+\mu^-$. The selections that each trigger line performs are as follows:

- *TrackMVA* requires a track with a minimum transverse momentum and a minimum significance of its impact parameter with respect to any primary vertex. These requirements are the same for the CPU- and GPU-based implementation, thus very similar performance is expected for this line.
- *TwoTrackMVA* combines two displaced tracks of good quality with a minimum momentum to a vertex. In the CPU-based implementation a multivariate analysis (MVA) classifier is trained to identify those vertices which originate from a decay of a long-lived particle such as a beauty or charmed hadron. An MVA is not yet in place for the GPU-based implementation, so percent level differences in the signal efficiencies are expected for a given background retention.

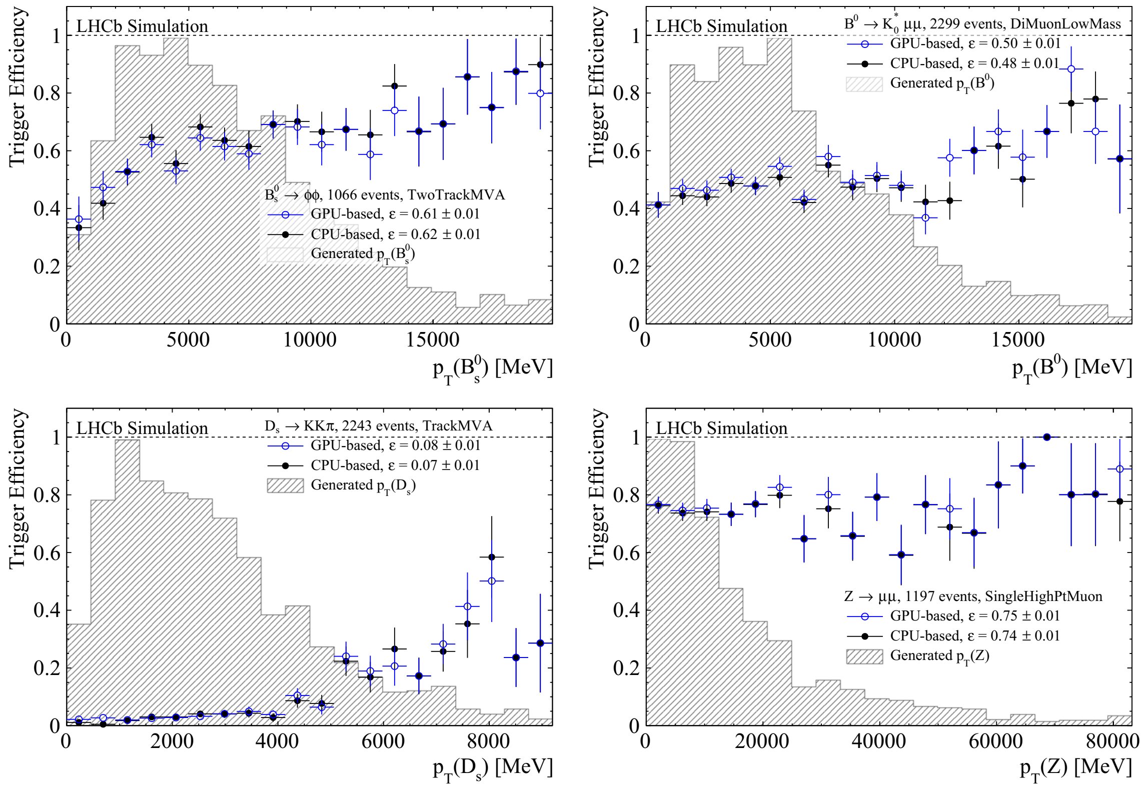

Fig. 11 Trigger efficiencies for CPU-based and GPU-based HLT1 as a function of parent transverse momentum. Results are shown for the `TwoTrackMVA` (top left), `DiMuonLowMass` (top right), `TrackMVA` (bottom left) and `SingleHighPtMuon` (bottom right) selections firing on the $B_s^0 \to \phi\phi$, $B^0 \to K^{*0}\mu^+\mu^-$, $D_s \to KK\pi$ and $Z \to \mu^+\mu^-$ signal samples, respectively. The generated parent transverse momentum distribution is also shown for all events passing the denominator requirement

- *DiMuonLowMass* requires a two track combination of displaced tracks with very low momentum and transverse momentum. This trigger line is implemented identically for the CPU- and GPU-based HLT1.
- *SingleHighPtMuon* selects tracks which are identified as a muon and fulfill a minimum momentum and transverse momentum requirement. This trigger line is also implemented identically for both architectures.

In Figs. 11 and 12 , the trigger efficiencies of the CPU- and GPU-based HLT1 implementations are plotted against the parent transverse momentum, p_T, and parent decay time, τ (where applicable), respectively, for four signal samples. For ease of interpretation, only the efficiency of one suitably chosen trigger line per sample is shown. The trigger efficiencies of the two implementations are found to be comparable.

In Tables 2 and 3 the trigger efficiencies, integrated across the kinematic phase space of the samples, are compared for all four selections and various simulated signal samples. They are found to be comparable for the GPU and CPU-based implementation.

HLT1 Rates

HLT1 rates are calculated in a similar way to the HLT1 efficiencies in the previous subsection. Both the CPU- and GPU-based implementation are run over 10 k minimum bias events, and the positive decisions on each of the four selections defined in the previous subsection are counted. The rate for each line is defined as the number of events that fire that line, divided by the number of minimum bias events that are sampled, multiplied by the LHCb non-empty bunch crossing rate (30 MHz). Note that, contrary to the efficiency studies on MC signal samples of the previous subsection, no preselection (or “denominator” requirement) is applied. The comparison of the HLT1 rates is shown in Fig. 13. The `TrackMVA`, `DiMuonLowMass` and `SingleHighPt-Muon` selections have comparable rates, although there is a discrepancy in the rates of the respective `TwoTrackMVA`

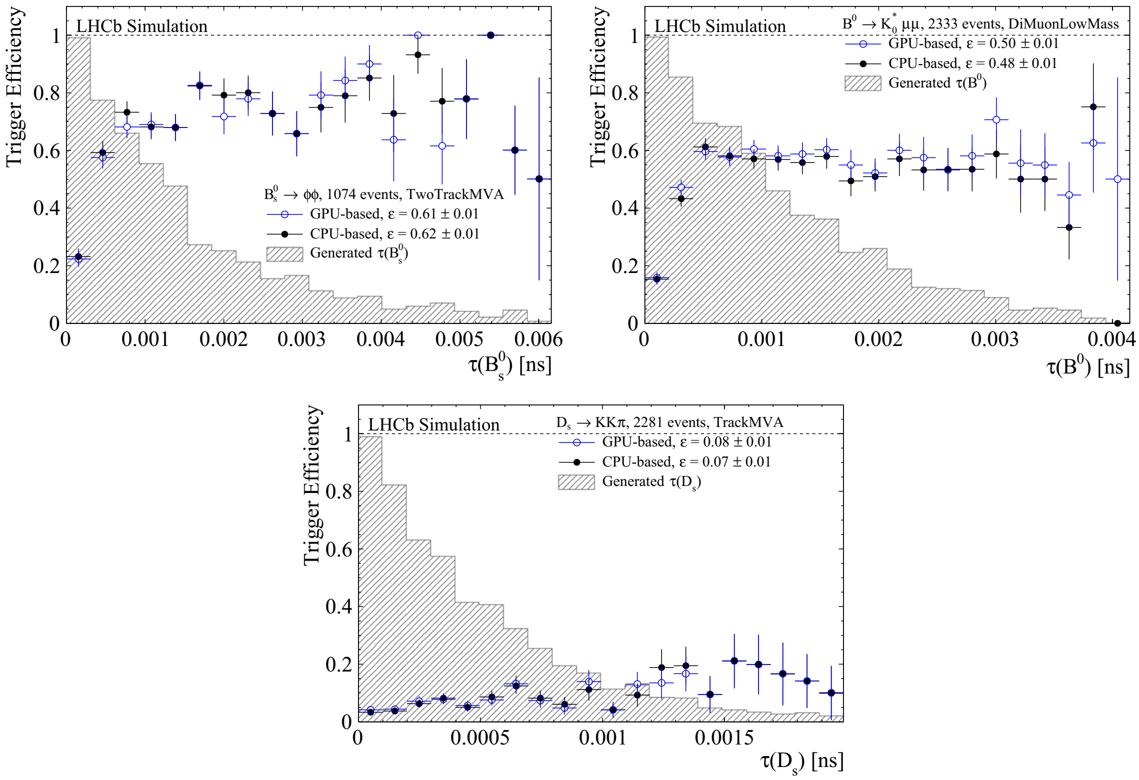

Fig. 12 Trigger efficiencies for CPU-based and GPU-based HLT1 as a function of parent decay time. Results are shown for the `TwoTrackMVA` (top left), `DiMuonLowMass` (top right) and `TrackMVA` (bottom) selections firing on the $B_s^0 \to \phi\phi$, $B^0 \to K^{*0}\mu^+\mu^-$ and $D_s \to KK\pi$ signal samples, respectively. The generated parent decay time distribution is also shown for all events passing the denominator requirement

Table 2 Comparison of trigger efficiencies integrated over the kinematic phase space of the candidates, for each of the six simulated signal samples for the `TrackMVA` and `TwoTrackMVA` selections

Signal	`TrackMVA`		`TwoTrackMVA`	
	GPU	CPU	GPU	CPU
$B_s^0 \to \phi\phi$	0.340(14)	0.332(14)	0.606(15)	0.621(15)
$J/\psi \to \mu^+\mu^-$	0.034(4)	0.031(3)	0.049(4)	0.042(4)
$B^0 \to K^{*0}e^+e^-$	0.276(10)	0.278(10)	0.439(12)	0.473(12)
$B^0 \to K^{*0}\mu^+\mu^-$	0.391(10)	0.385(10)	0.554(10)	0.582(10)
$D_s \to KK\pi$	0.076(5)	0.073(5)	0.178(8)	0.193(8)
$Z \to \mu^+\mu^-$	0.051(6)	0.040(6)	0.024(4)	0.028(5)

Statistical uncertainties are indicated in parentheses

Table 3 Comparison of trigger efficiencies integrated over the kinematic phase space of the candidates, for each of the six MC signal samples and the `DiMuonLowMass` and `SingleHighPtMuon` selections

Signal	`DiMuonLowMass`		`SingleHighPtMuon`	
	GPU	CPU	GPU	CPU
$B_s^0 \to \phi\phi$	0.025(5)	0.024(5)	0.005(2)	0.004(2)
$J/\psi \to \mu^+\mu^-$	0.078(5)	0.067(5)	0.048(4)	0.045(4)
$B^0 \to K^{*0}e^+e^-$	0.024(4)	0.027(4)	0.0011(8)	0.0011(8)
$B^0 \to K^{*0}\mu^+\mu^-$	0.502(10)	0.482(10)	0.091(6)	0.088(6)
$D_s \to KK\pi$	0.018(3)	0.019(3)	0.0013(7)	0.0013(7)
$Z \to \mu^+\mu^-$	0.033(5)	0.036(5)	0.749(12)	0.740(13)

Statistical uncertainties are indicated in parentheses

selections. This discrepancy can be explained by the different implementation of this line across the two projects, as detailed in the previous subsection. The inclusive rate for these four selections is found to be 912 ± 52 kHz for the GPU-based implementation, and 798 ± 48 kHz for the CPU-based one, largely due to the different implementation of the `TwoTrackMVA` trigger line. These numbers are well within the requirements of HLT1 to output between 1 and 2 MHz of events for further processing.

Global Event Cut Efficiency

Both HLT1 implementations apply an identical global event cut (GEC) requiring fewer than 9750 total SciFi and UT clusters. Consequently, the GEC efficiencies for the GPU- and CPU-based implementation are found to be identical

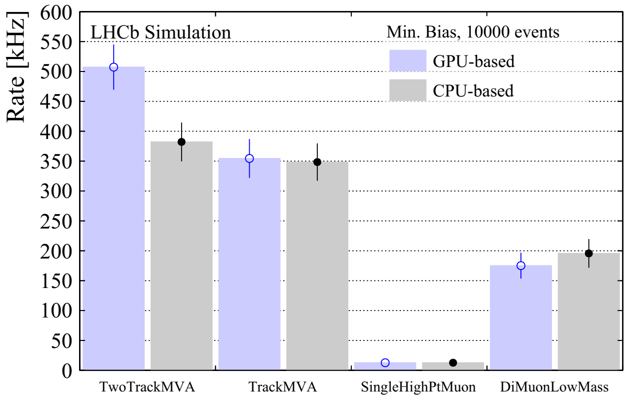

Fig. 13 Trigger rates of the CPU- and GPU-based implementation for the four trigger selections of interest. The difference in the rate of the TwoTrackMVA selections can be explained by the differing implementation, which is detailed in the "HLT1 Efficiency" section

across all samples. The integrated efficiency of $Z \to \mu^+\mu^-$ is found to be 0.75 ± 0.01. The other B and D decay samples under study have GEC efficiencies of about 85%, with statistical uncertainties of $\sim 1\%$. The efficiency on minimum bias events is 0.931 ± 0.003.

Summary of the Physics Performance

The GPU- and CPU-based implementations presented here result in close to identical performance in all aspects. The observed differences are considered to be a matter of tuning the algorithms to balance between efficiency and fake rate, or misidentification rate, and are expected to be neutral in terms of throughput. We therefore conclude that only the economic costs and the costs in terms of developer time need to be further considered in the cost–benefit calculation for the two architectures.

Cost–Benefit Analysis

In this section, the information of the previous sections is brought together to estimate the cost required for each architecture to process 30 MHz of data at the nominal instantaneous luminosity of $2 \times 10^{33}/\mathrm{cm}^2/\mathrm{s}$. As computing prices are in rapid flux, and any real purchase would be subject to tendering, we give here relative "costs" in units of the reference "Quanta" CPU server node used for the HLT during Run 2 data collection.

Nominal Cost–Benefit

The nominal cost–benefit analysis is based on the assumption that the LHC will run at 30 MHz with full luminosity from the start of data taking. It is summarized in Table 4.

Table 4 Indicative overall cost of the HLT1 implementations including contingency in units of the reference "Quanta" CPU server node used for the HLT during Run 2 data collection

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

Numbers have been rounded to reflect inevitable order (10%) fluctuations in real-world costs depending on the context of any given purchase

The difference in capital expenditure costs for the CPU-only and the hybrid scenario is the sum of the difference in HLT1 compute costs and the HLT1 network costs. For the HLT1 compute costs of the CPU-only scenario, it should be noted that additional usage of the EFF is foreseen, namely for processing simulation when the LHC is in a technical stop or end-of-year shutdown ("out-of-data-taking") and for processing HLT2 when the LHC is between fills ("out-of-fill"), respectively. While the out-of-fill use of the CPU nodes for HLT2 processing is taken into account in the following studies, the usage for simulation out-of-data taking is not.

Disk Buffer Simulation

Based on the boundary conditions outlined in Sect. 2, studies of the expected maximum HLT1 output rate that can be processed within the budget envelope have been performed. The studies are performed as follows:

1. fill lengths and inter-fill gaps are randomly sampled from 2018 data, with three machine development and technical stop periods and an average machine efficiency of 50%;
2. the fills and gaps are grouped into pairs before sampling to capture potential correlations between them;
3. a thousand such toys are randomly generated, and are shown to have a residual distribution compatible with that of the 2018 machine efficiency and a width of ±1.7%;
4. the disk buffer required to ensure sufficient space for 95% of these 1000 toys is then determined using as input an event size of 100 kB, a chosen HLT1 output rate, and a chosen HLT2 throughput in and out of fill;
5. these values are scanned, generating 1000 toys per data point, over a range of HLT1 output rates and HLT2 throughput rates resulting in a 3 dimensional distribu-

tion of required disk buffer as a function of HLT1 and HLT2 rates.

For each data point in the distribution, the cost of the HLT2 throughput and cost of the disk buffer are determined. For combined costs greater than that of the overall budget in the CPU-only and hybrid scenarios, the data point is rejected. This leaves a distribution of valid points for which LHCb could purchase the necessary resources. The optimal working point is the one which maximizes the HLT1 output rate. The inputs to this procedure are summarized in Table 6 and described in more detail in the following sections.

The simulation studies assume that in both the CPU-only and hybrid scenarios, any remaining budget after attaining a 30 MHz throughput at HLT1 and a sufficient buffer is used to buy CPU to provide additional HLT2 throughput. This corresponds to an increased HLT1 output rate allowing more, or more efficient, selections at HLT1.

Quantification of HLT1 and HLT2 Throughputs

The planned EFF nodes are equivalent to dual-socket AMD 7502 servers. The HLT1 throughput for this node was given in the “Throughput” section, requiring 175 nodes to sustain 30 MHz. As described in the “Throughput” section, these nodes are also capable of an HLT2 throughput of 471 Hz. The baseline scenario then results in an out-of-fill HLT2 throughput of 82.5 kHz and the remaining funds can be used to purchase disk or additional CPU resources. In the GPU scenario, the purchase of these nodes for HLT1 is not necessary, but as a result, no additional out-of-fill rate is available for HLT2 beyond the free resources described in the next subsection. The throughputs and quantities of these nodes before cost–benefit optimization are presented in Table 5.

Additional Free Computing Resources for HLT2

The Run 2 EFF Quanta nodes currently sustain a throughput of 134 Hz for HLT2, as documented in the “Throughput” section. In addition to the most recently purchased Run 2 EFF nodes, older nodes are available which correspond to 1450 quanta-equivalents, with up to 1200 additional nodes. These nodes will only be used for HLT2. In this study we take the 1450 nodes and assume a conservative 50% of the additional nodes, totalling 2050 quanta equivalents, or 275 kHz of in- and out-of-fill HLT2 throughput.

Results

The two scenarios are scanned according to the method described in Sect. 6.2. The results are shown in Fig. 14. In these figures, any combination of additional CPU and disk buffer that is cheaper than the total budget for the two scenarios is shown, defining a region of affordability, with the necessary disk buffer capacity represented by the z-axis colour. Unsurprisingly the maximum HLT1 throughput that can be sustained arises when the buffer is fully used and the remaining resources are spent exclusively on HLT2. The maximum HLT1 sustainable throughput in these scenarios is provided in Table 7, corresponding to the maximum y-axis extent of the region of affordability.

Table 5 Summary of HLT1 and HLT2 throughputs for the EFF nodes described in the “Throughput” section

Node type	HLT1 Throughput/node	HLT2 Throughput/node	Minimum nodes
AMD 7502	171 kHz	471 Hz	175
Quanta	n/a	134 Hz	2050

A minimum of 175 AMD 7502 nodes would be required to sustain 30 MHz of HLT1 throughput, corresponding to 82.5 kHz of out-of-fill HLT2 throughput. This is supplemented by 2050 Quanta equivalent nodes which are only used for HLT2 in-fill and out-of-fill, providing an additional 275 kHz of HLT2 throughput

Table 6 Input parameters to the toy studies used to determine the maximum HLT1 output rate in each scenario

Scenario	CPU-only	Hybrid
Maximum usable disk (PB)	28	28
In-fill min. HLT2 rate (kHz)	275	275
Out-of-fill min. HLT2 rate (kHz)	358	275

Table 7 Results of the toy studies indicating the maximum affordable throughput in both scenarios

Scenario	Maximum HLT1 output rate (kHz)
CPU-only	660
Hybrid	790

The disk buffer is used to its maximum of 28 PB

Results Assuming a Factor Two Performance Increase in HLT2

HLT2 has not been optimized to the same extent as HLT1; therefore, it is expected to improve its performance. To highlight the importance of doing so, the previous studies are repeated assuming a factor two improvement in the HLT2 throughput using the same architectures. This is motivated by preliminary studies using optimizations to the reconstruction sequence that are pending validation.

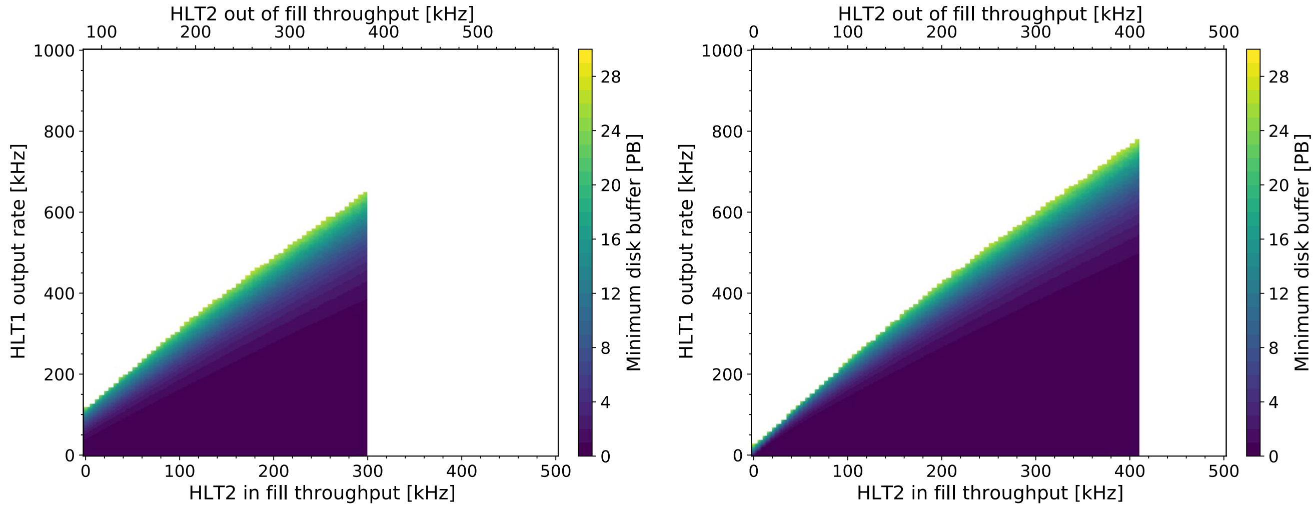

Fig. 14 Region of affordability for the CPU-only scenario (left) and the hybrid scenario (right). The maximum affordable HLT1 throughput is 660 kHz and 790 kHz, respectively

Table 8 Results of the toy studies indicating the maximum affordable throughput in the four scenarios in which HLT2 throughput has increased by a factor of two

Scenario	Maximum HLT1 output rate (kHz)
CPU-only	1185
Hybrid	1425

In each case, the maximum throughput arises when the full remaining spend is used on HLT2 and the disk buffer is used to its maximum of 28 PB respectively

The two scenarios with double the HLT2 performance result in the regions of affordability shown in Fig. 15. As before, the maximum HLT1 throughput that can be sustained arises when the buffer is fully used and the remaining resources are spent exclusively on HLT2. The maximum HLT1 sustainable throughput in these scenarios is provided in Table 8.

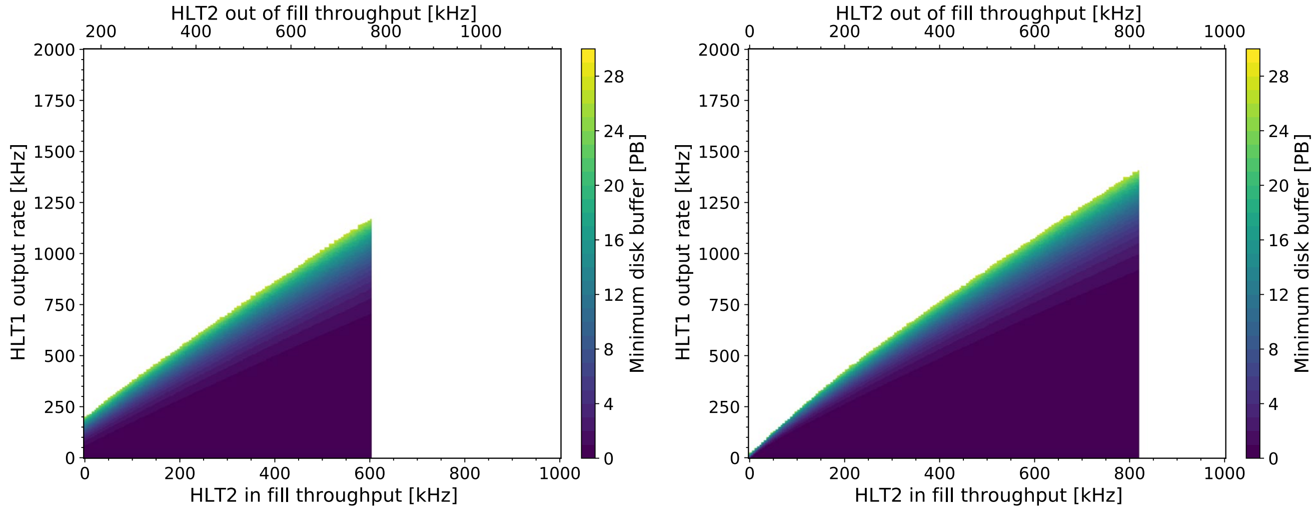

Fig. 15 Region of affordability for the CPU-only scenario (left) and the hybrid scenario (right) when HLT2 throughput is twice as high as in the current best performance. The maximum affordable HLT1 throughput is 1185 kHz and 1425 kHz, respectively

Deferred Purchasing

It is expected that the first year of Run 3 will be a commissioning year, with stable pp running at lower than nominal luminosities early in the year. This is expected to be followed by a period of running at nominal luminosity. There is general agreement that, if possible, hardware purchasing should be deferred to make best use of the collaboration's financial resources. For this reason, the plan is to buy a system for 2022 which can handle half the expected nominal processing load. As the throughput of both the considered HLT1 architectures scales linearly (or worse) with detector occupancy, this implies that buying half the number of HLT1 processing units is sufficient. Many of the relevant costs from Table 4 can therefore be divided by two.

In the case of the CPU-only architecture, the disk storage required for 1 MHz output involves a great deal of additional mechanical and cabling work if purchased in two parts, so it is preferable to buy the full amount already for the commissioning year. However, it is also technically possible to divide this purchase, in case there are good financial reasons to expect that it saves the collaboration a significant amount of money. Similar arguments also apply to the network between the EB and the CPU HLT1.

In the hybrid case, deferred purchasing allows HLT1 to be implemented with one GPU card per EB node in the first year. This results in maximum flexibility to benefit from the next generation of GPU cards which are expected to be released over the next 2 years, and mitigates the risk arising from the limited number of EB slots available to implement the GPU HLT1.

Conclusion

LHCb has investigated two viable proposals for its first-level trigger to be deployed for Run 3 onwards. They are characterized by similar performance: the choice between x86-based and GPU-based technologies is far from obvious when viewed solely in terms of raw throughput and physics performance, and as a result, the decision over which option to use has come down to a detailed cost/benefit assessment, presented in this paper.

In addition to certain immediate cost savings, the GPU-based solution offers more opportunities for future performance gains, as a more forward-looking longer-term investment; it is anticipated that GPU performance will advance at a more rapid pace, as evidenced by the new generation of GPU cards released while this paper was being prepared. This increased growth rate presents an opportunity to do more physics in Run 3 and beyond; and provides LHCb with the expertise and increased flexibility to exploit fully heterogeneous computing environments in future upgrades. The overhead of maintaining different technologies on HLT1 and HLT2 is not negligible and has been taken into account in the decision making process as well. On balance, however, as a result of all these considerations, the collaboration has ultimately decided to implement a hybrid HLT with a GPU-based first stage, in Run 3. The studies presented in this paper show that GPUs are increasingly a viable alternative to CPUs as general-purpose, quasi-standalone processors.

Acknowledgements 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); MEiN and NCN UMO-2018/31/B/ST2/03998 (Poland); MEN/IFA (Romania); MSHE (Russia); MICINN (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 NERSC (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 ARC and ARDC (Australia); AvH Foundation (Germany); EPLANET, Marie Skłodowska-Curie Actions and ERC (European Union); A*MIDEX, ANR, IPhU and Labex P2IO, and Région Auvergne-Rhône-Alpes (France); Key Research Program of Frontier Sciences of CAS, CAS PIFI, CAS CCEPP, Fundamental Research Funds for the Central Universities, and Sci. & Tech. Program of Guangzhou (China); RFBR, RSF and Yandex LLC (Russia); GVA, XuntaGal and GENCAT (Spain); the Leverhulme Trust, the Royal Society and UKRI (United Kingdom). The authors would like to thank the LHCb computing and simulation teams for their support and for producing the simulated LHCb samples used to benchmark the performance of RTA software.

Funding Open access funding provided by CERN (European Organization for Nuclear Research).

Data Availability Statement The data sets generated during and/or analysed during the current study are not publicly available as LHCb collaboration policy is to make data fully available 5 years after it has been analysed. They are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest No author of this paper has a financial or personal relationship with a third party whose interests could be positively or negatively influenced by the article's content.

References

1. LHCb collaboration, Alves Jr AA et al (2008) The LHCb detector at the LHC. JINST 3:S08005, https://doi.org/10.1088/1748-0221/3/08/S08005
2. Aaij R et al (2019) Performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. JINST 14:P04013. https://doi.org/10.1088/1748-0221/14/04/P04013. arXiv:1812.10790
3. Akar S et al (2014) Review document: low level trigger (LLT). CERN, Geneva
4. ATLAS Collaboration (2017) Technical design report for the phase-II upgrade of the ATLAS TDAQ system. CERN, Geneva
5. CMS Collaboration (2017) The Phase-2 Upgrade of the CMS DAQ Interim Technical Design Report, CERN, Geneva. This is the CMS Interim TDR devoted to the upgrade of the CMS DAQ in view of the HL-LHC running, as approved by the LHCC
6. Buncic P, Krzewicki M, Vande Vyvre P (2015) Technical design report for the upgrade of the online-offline computing system
7. LHCb collaboration (2014) lhcb trigger and online technical design report, CERN-LHCC-2014-016. http://cdsweb.cern.ch/search?p=CERN-LHCC-2014-016&f=reportnumber&action_search=Search&c=LHCb+Reports. Accessed 28 Dec 2021
8. Aaij R et al (2020) Allen: a high level trigger on GPUs for LHCb. Comput Softw Big Sci 4:7. https://doi.org/10.1007/s41781-020-00039-7. arXiv:1912.09161
9. LHCb collaboration (2020) LHCb upgrade GPU high level trigger technical design report, CERN-LHCC-2020-006. http://cdsweb.cern.ch/search?p=CERN-LHCC-2020-006&f=reportnumber&action_search=Search&c=LHCb+Reports. Accessed 28 Dec 2021
10. for the ALICE Collaboration, Gorbunov S et al (2011) ALICE HLT high speed tracking on GPU. IEEE Trans Nucl Sci 58: 1845, https://doi.org/10.1109/TNS.2011.2157702
11. LHCb Collaboration (2020) Performance of the GPU HLT1 (Allen), LHCB-FIGURE-2020-014. http://cdsweb.cern.ch/search?p=LHCB-FIGURE-2020-014&f=reportnumber&action_search=Search&c=LHCb+Figures. Accessed 28 Dec 2021
12. Aaij R et al (2016) Tesla: an application for real-time data analysis in High Energy Physics. Comput Phys Commun 208:35. https://doi.org/10.1016/j.cpc.2016.07.022. arXiv:1604.05596
13. Aaij R et al (2019) A comprehensive real-time analysis model at the LHCb experiment. JINST 14:P04006. https://doi.org/10.1088/1748-0221/14/04/P04006. arXiv:1903.01360
14. LHCb collaboration (2018) Computing model of the upgrade LHCb experiment, CERN-LHCC-2018-014. http://cdsweb.cern.ch/search?p=CERN-LHCC-2018-014&f=reportnumber&action_search=Search&c=LHCb+Reports. Accessed 28 Dec 2021

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

R. Aaij[21] · M. Adinolfi[34] · S. Aiola[17] · S. Akar[43] · J. Albrecht[10] · M. Alexander[39] · S. Amato[1] · Y. Amhis[8] · F. Archilli[12] · M. Bala[25] · G. Bassi[20,55] · L. Bian[45] · M. P. Blago[31] · T. Boettcher[42] · A. Boldyrev[48] · S. Borghi[40] · A. Brea Rodriguez[29] · L. Calefice[9,10] · M. Calvo Gomez[49] · D. H. Cámpora Pérez[31,47] · A. Cardini[19] · M. Cattaneo[31] · V. Chobanova[29] · G. Ciezarek[31] · X. Cid Vidal[29] · J. L. Cobbledick[40] · J. A. B. Coelho[8] · T. Colombo[31] · A. Contu[19] · B. Couturier[31] · D. C. Craik[42] · R. Currie[38] · P. d'Argent[31] · M. De Cian[32] · D. Derkach[48] · F. Dordei[19] · M. Dorigo[20,56] · L. Dufour[31] · P. Durante[31] · A. Dziurda[24] · A. Dzyuba[26] · S. Easo[37] · S. Esen[9] · P. Fernandez Declara[31] · S. Filippov[27] · C. Fitzpatrick[40] · M. Frank[31] · P. Gandini[17] · V. V. Gligorov[9] · E. Golobardes[49] · G. Graziani[14] · L. Grillo[40] · P. A. Günther[12] · S. Hansmann-Menzemer[12] · A. M. Hennequin[31] · L. Henry[17,30] · D. Hill[32] · S. E. Hollitt[10] · J. Hu[12] · W. Hulsbergen[21] · R. J. Hunter[36] · M. Hushchyn[48] · B. K. Jashal[30] · C. R. Jones[35] · S. Klaver[21] · K. Klimaszewski[25] · R. Kopecna[12] · W. Krzemien[25] · M. Kucharczyk[24] · R. Lane[34] · F. Lazzari[20,55] · R. Le Gac[7] · P. Li[12] · J. H. Lopes[1] · M. Lucio Martinez[21] · A. Lupato[40] · O. Lupton[36] · X. Lyu[4] · F. Machefert[8] · O. Madejczyk[23] · S. Malde[41] · J. F. Marchand[6] · S. Mariani[14,31,51] · C. Marin Benito[31] · D. Martinez Santos[29] · F. Martinez Vidal[30] · R. Matev[31] · M. Mazurek[31] · B. Mitreska[40] · D. S. Mitzel[31] · M. J. Morello[20,53] · H. Mu[2] · P. Muzzetto[19,31] · P. Naik[34] · M. Needham[38] · N. Neri[17,52] · N. Neufeld[31] · N. S. Nolte[10,31] · D. O'Hanlon[34] · A. Oyanguren[30] · M. Pepe Altarelli[31] · S. Petrucci[38] · M. Petruzzo[17] · L. Pica[20,55] · F. Pisani[31] · A. Piucci[12] · F. Polci[9] · A. Poluektov[7] · E. Polycarpo[1] · C. Prouve[29] · G. Punzi[20,54] · R. Quagliani[9] · R. I. Rabadan Trejo[7] · M. Ramos Pernas[36] · M. S. Rangel[1] · F. Ratnikov[41,48] · G. Raven[22] · F. Reiss[9] · V. Renaudin[41] · P. Robbe[8] · A. Ryzhikov[48] · M. Santimaria[15] · M. Saur[10] · M. Schiller[39] · R. Schwemmer[31] · B. Sciascia[15] · A. Solomin[34,50] · F. Suljik[41] · N. Skidmore[40] · M. D. Sokoloff[43] · P. Spradlin[39] · M. Stahl[43] · S. Stahl[31] · H. Stevens[10] · L. Sun[45] · A. Szabelski[25] · T. Szumlak[23] · M. Szymanski[31] · D. Y. Tou[9] · G. Tuci[20,54] · A. Usachov[21] · N. Valls Canudas[28] · R. Vazquez Gomez[29] · S. Vecchi[13] · M. Vesterinen[36] · X. Vilasis-Cardona[49] · D. Vom Bruch[7] · Z. Wang[33] · T. Wojton[24] · M. Whitehead[34] · M. Williams[42,44] · M. Witek[24] · Y. Xie[5] · A. Xu[3] · H. Yin[5] · M. Zdybal[24] · O. Zenaiev[31] · D. Zhang[5] · L. Zhang[2] · X. Zhu[2] · The LHCb Collaboration[11,16,18,46]

✉ V. V. Gligorov
vladimir.gligorov@cern.ch

[1] Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

[2] Center for High Energy Physics, Tsinghua University, Beijing, China

[3] School of Physics State Key Laboratory of Nuclear Physics and Technology, Peking University, Beijing, China

[4] University of Chinese Academy of Sciences, Beijing, China

[5] Institute of Particle Physics, Central China Normal University, Wuhan, Hubei, China

[6] Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, IN2P3-LAPP, Annecy, France

[7] Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France

[8] Université Paris-Saclay, CNRS/IN2P3, IJCLab, Orsay, France

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

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

[11] Max-Planck-Institut für Kernphysik (MPIK), Heidelberg, Germany

[12] Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany

[13] INFN Sezione di Ferrara, Ferrara, Italy

[14] INFN Sezione di Firenze, Florence, Italy

[15] INFN Laboratori Nazionali di Frascati, Frascati, Italy

[16] INFN Sezione di Genova, Genoa, Italy

[17] INFN Sezione di Milano, Milan, Italy

[18] INFN Sezione di Milano-Bicocca, Milan, Italy

[19] INFN Sezione di Cagliari, Monserrato, Italy

[20] INFN Sezione di Pisa, Pisa, Italy

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

[22] Nikhef National Institute for Subatomic Physics and VU University Amsterdam, Amsterdam, The Netherlands

[23] Faculty of Physics and Applied Computer Science, AGH, University of Science and Technology, Kraków, Poland

[24] Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences, Kraków, Poland

[25] National Center for Nuclear Research (NCBJ), Warsaw, Poland

[26] Petersburg Nuclear Physics Institute NRC Kurchatov Institute (PNPI NRC KI), Gatchina, Russia

[27] Institute for Nuclear Research of the Russian Academy of Sciences (INR RAS), Moscow, Russia

[28] ICCUB, Universitat de Barcelona, Barcelona, Spain

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

[30] Instituto de Fisica Corpuscular, Centro Mixto Universidad de Valencia, CSIC, Valencia, Spain

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

[32] Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

[33] Physik-Institut, Universität Zürich, Zürich, Switzerland

[34] H.H. Wills Physics Laboratory, University of Bristol, Bristol, UK

[35] Cavendish Laboratory, University of Cambridge, Cambridge, UK

[36] Department of Physics, University of Warwick, Coventry, UK

[37] STFC Rutherford Appleton Laboratory, Didcot, UK

[38] School of Physics and Astronomy, University of Edinburgh, Edinburgh, UK

[39] School of Physics and Astronomy, University of Glasgow, Glasgow, UK

[40] Department of Physics and Astronomy, University of Manchester, Manchester, UK

[41] Department of Physics, University of Oxford, Oxford, UK

[42] Massachusetts Institute of Technology, Cambridge, MA, USA

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

[44] School of Physics and Astronomy, Monash University, Melbourne, Australia

[45] School of Physics and Technology, Wuhan University, Wuhan, China

[46] Universität Bonn, Helmholtz-Institut für Strahlen und Kernphysik, Bonn, Germany

[47] Universiteit Maastricht, Maastricht, The Netherlands

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

[49] DS4DS, La Salle, Universitat Ramon Llull, Barcelona, Spain

[50] Institute of Nuclear Physics, Moscow State University (SINP MSU), Moscow, Russia

[51] Università di Firenze, Florence, Italy

[52] Università degli Studi di Milano, Milan, Italy

[53] Scuola Normale Superiore, Pisa, Italy

[54] Università di Pisa, Pisa, Italy

[55] Università di Siena, Siena, Italy

[56] INFN Sezione di Trieste, Trieste, Italy