

# Monitoring reconstruction software in LHCb

Ying-Rui Hou<sup>1,\*</sup>, Suzanne Klaver<sup>2,\*\*</sup>, Sneha Malde<sup>3,\*\*\*</sup>, Rosen Matev<sup>4,\*\*\*\*</sup>, Dmitry Popov<sup>1,†</sup>, and Miroslav Saur<sup>5,‡</sup>

<sup>1</sup>University of Chinese Academy of Sciences, Beijing, China

<sup>2</sup>Nikhef National Institute for Subatomic Physics, Amsterdam, Netherlands

<sup>3</sup>University of Oxford, Oxford, United Kingdom

<sup>4</sup>European Organization for Nuclear Research (CERN), Geneva, Switzerland

<sup>5</sup>Technische Universität Dortmund, Dortmund, Germany

**Abstract.** The LHCb detector at the LHC is currently undergoing a major upgrade to increase its full detector read-out rate to 30 MHz. In addition to the detector hardware modernisation, the new trigger system will be software-only. The code base of the new trigger system must be thoroughly tested for data flow, functionality and physics performance. Currently, the testing procedure is based on a system of nightly builds and continuous integration tests of each new code development. The continuous integration tests are now extended to test and evaluate high-level quantities related to LHCb's physics program, such as track reconstruction and particle identification, which is described in this paper. Before each merge request, the differences after the change in code are shown and automatically compared using an interactive visualisation tool, allowing easy verification of all relevant quantities. This approach gives an extensive control over the physics performance of the new code resulting into better preparation for data taking with the upgraded LHCb detector at Run 3.

## 1 Introduction

The LHCb experiment located at the LHC at CERN has collected data from 2010-2012 (Run 1), and 2015-2018 (Run 2). At the moment the experiment is undergoing a major upgrade to prepare it for the Run 3 period from 2022-2024. Run 3 is envisaged to have a five times higher instantaneous luminosity as during Run 1 and 2 [1]. In order to maintain performance at this higher instantaneous luminosity, the majority of the detector sub-systems are being upgraded and replaced. In addition, the trigger system is entirely restructured and the hardware trigger that used to reduce the data rate to 1 MHz, and limited the efficiency of key parts of the scientific programme, has been removed.

A key part of the upgrade is the development of the full software trigger processing data at a collision rate of 30 MHz. The trigger itself is divided into two stages, the High Level

---

\*e-mail: [yingrui.hou@cern.ch](mailto:yingrui.hou@cern.ch)

\*\*e-mail: [suzanne.klaver@cern.ch](mailto:suzanne.klaver@cern.ch)

\*\*\*e-mail: [sneha.malde@physics.ox.ac.uk](mailto:sneha.malde@physics.ox.ac.uk)

\*\*\*\*e-mail: [rosen.matev@cern.ch](mailto:rosen.matev@cern.ch)

†e-mail: [dmitry.popov@cern.ch](mailto:dmitry.popov@cern.ch)

‡e-mail: [miroslav.saur@cern.ch](mailto:miroslav.saur@cern.ch)

Trigger 1 (HLT1) and the High Level Trigger 2 (HLT2). The HLT1 stage is a fully GPU-based implementation with a maximum input data rate of 5 TB/s and a maximum output rate of 0.25 TB/s [2]. Events at this stage are mainly selected using inclusive one- and two-track algorithms.

All the data passing the HLT1 selection will be fully reconstructed by the HLT2 sequence. This includes a real-time calibration and alignment of the detector which enables more stringent requirements on the data to be placed. The expected output from the trigger system is up to 10 GB/s with a typical event size of 100 kB [3]. The chosen trigger architecture offers substantial flexibility in changing the trigger conditions during the data-taking period and requires a well-tested code base with a fully understood physics performance. This need is even stronger since most of the detector itself will be replaced. This paper explains the test procedures that are currently in place in Sec. 2, and those that are under development in Sec. 3. A summary and further ideas are presented in Sec. 4.

## 2 Development and testing of the upgrade code base

The LHCb software stack is a highly modular system based on the Gaudi framework [4] consisting of separate projects where each project comprises libraries and applications which may have common dependencies. All the code is hosted on the CERN GitLab system and is described in detail in Ref. [5]. The modularity of the stack allows concurrent development of separate parts of the code base, and new code is submitted as a Merge Request (MR) to the corresponding projects. However, the drawback of this architecture is that the impact of the code changes on the rest of the stack can be hidden; *i.e.* a change in the main framework can interfere with the functionality of specific projects, libraries or applications.

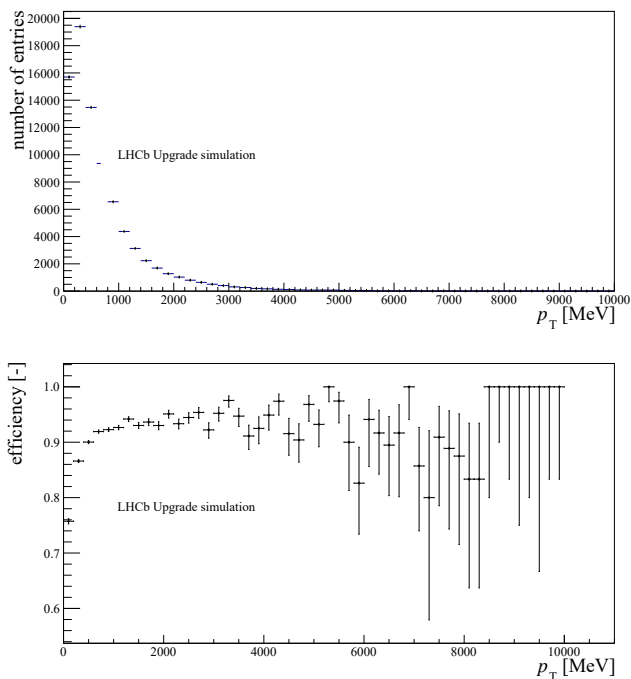
To keep track of the interference between different parts of the code, the LHCb software stack must be tested thoroughly on a regular basis with feedback that is easily available to all developers. The testing infrastructure at LHCb is based around the system of nightly builds [6] and has evolved significantly in the past years, as presented in previous CHEP contributions. Continuous Integration (CI) methods have been made more flexible using web-hooks [7]. In addition, the LHCb Performance Regression (LHCbPR) framework [8] and subsequent improvements focusing on usage of message queues [9] have improved the testing capabilities.

The LHCb nightly build system consists of approximately 300 centrally managed cores. The full stack is periodically built and tested on different platforms, where platform is defined as a combination of a specific operating system, hardware architecture and compiler. Originally this procedure was triggered only by a timer and ran daily. Under these settings any debugging takes a significant amount of time, as it takes at least 24 hours to locate the problem and test a possible fix. Later, a system of web-hooks was developed, allowing more flexible triggering. This system can trigger an isolated test of the newly developed code and its impact on the stack at any time, thus significantly reducing feedback time. Two instances of the stack are compiled, tested and compared, one corresponding to the master version of the stack and another, which includes changes from the MR. The nightly builds and more flexible testing are using the same hardware infrastructure. The main aim of both systems is to test that all of the software can be built, ran and finalised successfully without any compilation error, missing dependencies or other unexpected errors. The system of nightly builds is still evolving and is described in more details in [6, 7].

The LHCbPR framework is designed to test higher-level quantities including a wide spectrum of physics observables, ranging from basic distributions, such as the momentum of a particle, to more complicated tasks, for example the evaluation of tracking efficiencies. The PR tests are set up using configuration scripts and allow for a high flexibility in calling and

running selected projects within the full LHCb software stack. Due to the size of the samples that is run over in PR tests, their typical running time is longer than that for the relatively simple tests of the code functionality in nightly builds. The results produced by the PR tests are centrally stored and can be accessed for direct evaluation or used for further evaluation using the LHCbPR front-end web application. The LHCbPR framework is using the same nightly build system for running tests. The exact running time of the PR tests depends on the algorithms tested and, if needed, the size of the produced or input data samples. A typical running time is in the order of ten minutes.

The LHCbPR system, and in particular its web front-end, were initially developed for the needs of LHCb simulation group as a tool to quickly check new versions of the LHCb simulation software [10], and is currently also widely used for developing the trigger reconstruction software, allowing to aggregate results previously done by a separate groups of developers. An important part of the LHCbPR front-end is a dashboard which allows to automatically plot predefined sets of results, typically histograms, based on a selected LHCbPR test. This can be done for different platforms and compares results to the nightly builds from previous seven days. Additionally, specific references can be set up to mark an important result. A typical example of a reference is a new software release version. Any later builds can then be compared to this reference.



**Figure 1.** Example of tested observables. Top: transverse momentum of a  $B$  meson. Bottom: long track reconstruction efficiency.

### 3 Physics performance testing

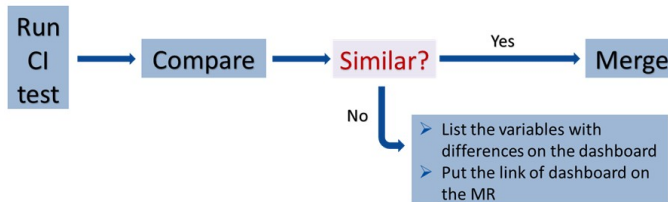
The nightly tests described above focus on the basic functionality of the code, such as compilation and running. However, to understand the full impact of the new code and its perfor-

mance, more thorough and complex tests are needed. This can be achieved by monitoring changes in the physics performance as a result of a new MR to the master version of the software stack using a predefined set of tests which cover most of the use cases at LHCb.

Based on the work described in Sec. 2 we developed a new set of centrally executed physics performance tests as part of the LHCbPR framework. Such automated tests did not yet exist or were tested only by a relevant group of developers. These tests focus on the reconstruction part of LHCb’s software trigger. In order to cover the main characteristics of LHCb’s broad physics programme, ten distinct decay modes using the nominal upgrade parameters are simulated:  $Z \rightarrow \mu^+ \mu^{-1}$ ,  $Z \rightarrow e^+ e^-$ ,  $B_s^0 \rightarrow J/\psi(\rightarrow \mu^+ \mu^-) \phi$ ,  $B_s^0 \rightarrow \phi \phi$ ,  $B^0 \rightarrow K^{*0} e^+ e^-$ ,  $B^0 \rightarrow K^{*0} \gamma$ ,  $B^0 \rightarrow D^{*-} \mu^+ \nu_\mu$ ,  $D^{*+} \rightarrow D^0(\rightarrow K^- \pi^+) \pi^+$ ,  $D^+ \rightarrow K_S^0 \pi^+$ , and minimum bias. The results shown in this paper are based on the  $B_s^0 \rightarrow \phi \phi$  samples.

In the simulation,  $pp$  collisions are generated using PYTHIA [11, 12] with a specific LHCb configuration [13]. Decays of unstable particles are processed by EVTGEN [14]. The interaction of the generated particles with the detector, and its response, are implemented using the GEANT4 toolkit [15, 16] as described in Ref. [17]. For the studies reported in this paper we use digitised simulated samples before reconstruction. Each simulated sample contains  $1 \times 10^5$  events, however only 1000 events are used for the nightly tests, as this size of simulated sample provides enough statistics for the reconstruction test without unnecessary additional stressing of the testing infrastructure and extending the required time. Since the simulated samples contain the detector response before the trigger stage, they can be used to test any changes in the reconstruction and selection stage. For example, after speeding up a track reconstruction algorithm, they are used to test the impact on how efficient the updated code is at reconstructing tracks.

The studied observables can be divided into the following categories: reconstructed primary vertex information, basic track properties (number and type of reconstructed tracks, track fit quality), kinematic variables (momentum and momentum transverse to the beam axis, pseudorapidity, azimuthal angle with respect to the beam axis), particle identification variables and track reconstruction efficiencies. An example of a typical transverse momentum,  $p_T$ , distribution and long track reconstruction efficiency as function of  $p_T$  is shown in Fig. 1. A long track is defined as the track with information from all LHCb tracking sub-detectors [18].

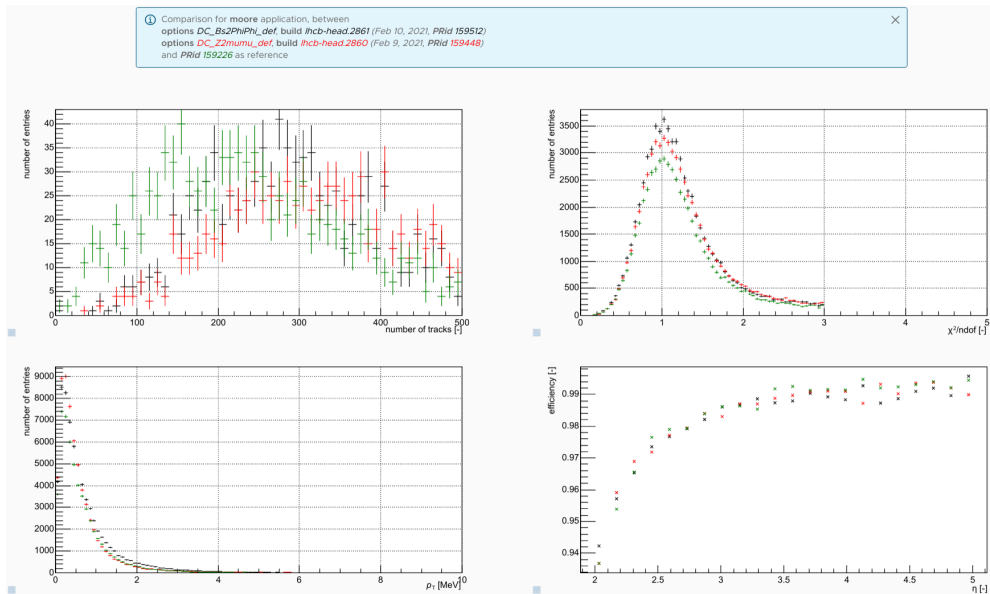


**Figure 2.** Proposed workflow for the new continuous-integration test.

The PR tests are run automatically every day and the selected set of observables is made visible on a dashboard to simplify the comparison between reconstruction software on different days. In addition, a more elaborate CI test is added to check before each MR if the code changes induce changes in performance. These tests are triggered using the same system of web-hooks and infrastructure as the CI test described before. The order of the test procedure is as follows: at first the master version of the code and the version where new MR

<sup>1</sup>The inclusion of charge-conjugate processes is implied throughout.

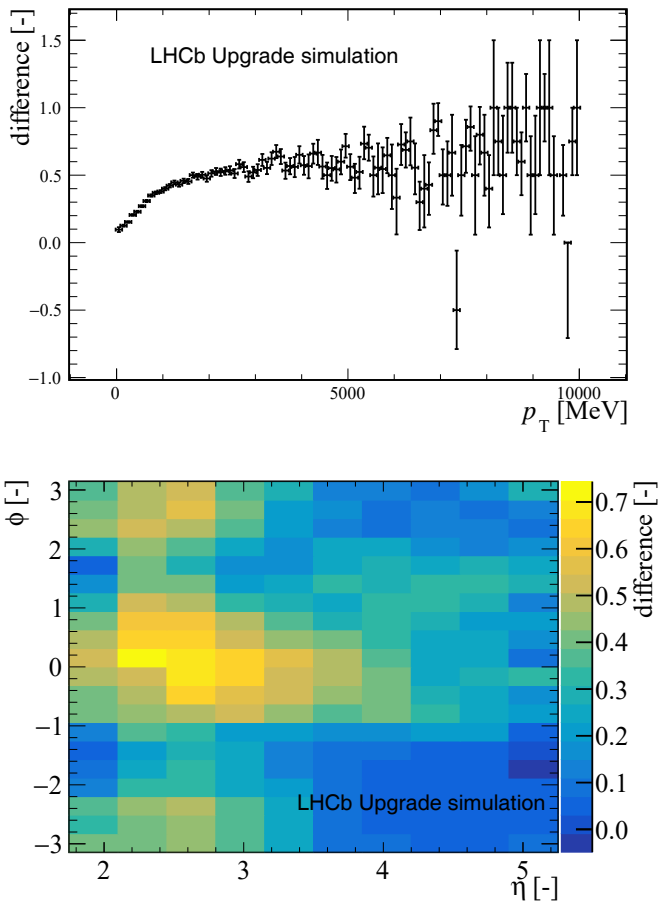
is merged into the code (MR version) are compiled. This allows variables for both versions to be checked independently. In addition, internal tests compare counters to check if the MR is ready to be merged. Secondly, the same set of physics performance tests is performed both on the master code and MR version. Finally, the test results from the separate builds are compared. In the case when the difference between the two builds is above a predefined threshold is observed, a message is sent to the MR GitLab page. This message contains a list of variables which are above the threshold and link to the web interface where all variables are visualised. In addition, the results are stored in the form of a ROOT file [19] within the LHCbPR framework for further studies if required. The workflow of these new tests is shown in Fig. 2 and an example of two observables visualised using the dashboard is shown in Fig. 3.



**Figure 3.** Screenshot of the reconstruction dashboard, showing (top left) the number of tracks per event, (top right) the track fit  $\chi^2/ndf$ , (bottom left) the  $p_T$  distribution and (bottom right) the track efficiency as a function of pseudorapidity  $\eta$ . For illustrative purposes different decays are compared instead of different builds. The green points represent the minimum bias sample, the red ones  $Z \rightarrow \mu^+\mu^-$  decays, and in black  $B_s^0 \rightarrow \phi\phi$  decays are shown.

The difference between two histograms is calculated as the sum of the absolute value of the difference in each bin. The threshold of this difference can differ per histogram, as some observables are expected to change frequently, while others should not change at all. In addition to that, two different lists are generated. The first is meant for MRs that should not affect any distribution at all. This list thus contains all histograms that show any difference. The second list takes into account small expected changes; the thresholds for most histograms are larger than zero.

Once the lists are available, developers and reviewers of the MR must evaluate the observed discrepancies by following the published links. The dashboard automatically generates a new web page where the flagged histograms and their differences are shown for additional checks. Examples of the difference between 1D and 2D histograms are shown in Fig. 4. The differences in reconstruction are exaggerated here to make them more visible.



**Figure 4.** Example of tested variables. Top: difference in the long track reconstruction efficiency between two different builds as a function of transverse momentum. Bottom: difference in the long track reconstruction efficiency as a function of pseudorapidity  $\eta$  and azimuthal angle  $\phi$ .

The results of the tests outlined above are available to all developers of the new software trigger and subsequently to all members of the LHCb collaboration. This allows an improved access to all results and decrease the reaction time to evaluate the impact of newly developed code. One example of the impact of the new system was shown when evaluating the effect of samples simulated by a new version of the simulation framework in which the descriptions of certain subdetectors were update. The developed LHCbPR dashboard served as a central place where all relevant distributions were plotted and could easily be compared with the results based on the previous simulated samples. This streamlined evaluating of the new subdetectors descriptions of the reconstruction code and eased finding differences.

## 4 Summary and outlook

To extend the testing capacity during the development of the new full software trigger at LHCb a set of tools for additional testing and monitoring based on the existing system of the

nightly tests and LHCbPR framework has been developed. This includes setting up new tests within the LHCbPR framework, preparing dedicated handlers for physics performance testing and expanding the LHCbPR web front-end in the form of new dashboards. Automated checks of the differences between physics observables produced by two builds of the LHCb software stack, allow to test a broader functionality of the full software trigger during the development stage. This check can for example be performed to test the current master version of the software stack and a version in which a new merge request is included and will become a part of the standard evaluation procedure of any merge request to projects which may influence the physics output.

The system discussed so far is implemented independently for the HLT1 and HLT2 stage of the trigger system, where both stages are run separately. The next step is to implement the same set of tests for the linked HLT1 and HLT2 system and subsequently for other trigger-related projects as they achieve the required readiness. This system already proved to be useful when evaluating the effect of new simulation samples on the reconstruction code. The main goal, to be achieved during the second half of 2021, is to set up a testing framework which will be able to test all the parts of the software trigger together and its performance.

In addition to this, more elaborate tests will be set up to monitor higher-level observables. This could for example be a mass distribution, for which more than 1000 events are needed. These types of variables should be monitored closely, but it is not necessary to do this for each MR. Therefore these will be monitored in a different framework. Ultimately, an ideal test would be indistinguishable from the real data-taking conditions, where the full software chain would be executed and its performance subsequently evaluated. These types of tests will be performed at the end of 2021 to test the readiness of the software.

## Acknowledgements

The authors would like to thank the LHCb computing and simulation teams for their support and for producing the simulated LHCb samples used in the paper. We would also like to thank the full RTA team for their valuable inputs to this project.

## References

- [1] LHCb Collaboration, Tech. rep. (2012), <https://cds.cern.ch/record/1443882>
- [2] LHCb Collaboration, Tech. rep., CERN, Geneva (2020), <https://cds.cern.ch/record/2717938>
- [3] LHCb Collaboration, Tech. rep., CERN, Geneva (2018), <https://cds.cern.ch/record/2310827>
- [4] G. Barrand, I. Belyaev, P. Binko, M. Cattaneo, R. Chytraccek, G. Corti, M. Frank, G. Gracia, J. Harvey, E. Herwijnen et al., *Computer Physics Communications* **140**, 45 (2001), CHEP2000
- [5] G. Corti, G. Barrand, I. Belyaev, M. Cattaneo, P. Charpentier, M. Frank, P. Koppenburg, P. Mato-Vila, F. Ranjard, S. Roiser, *IEEE Trans. Nucl. Sci.* **53**, 1323 (2006)
- [6] M. Clemencic, B. Couturier, *J. Phys.: Conf. Ser.* **513**, 052007. 5 p (2014)
- [7] S.G. Chitic, B. Couturier, M. Clemencic, J. Closier (LHCb Collaboration), *EPJ Web Conf.* **214**, 05001 (2019)
- [8] R. Currie, C. Fitzpatrick (LHCb Collaboration), *EPJ Web Conf.* **214**, 05042. 4 p (2019)
- [9] M. Szymański, B. Couturier (LHCb Collaboration), *EPJ Web Conf.* **214**, 05014. 7 p (2019)
- [10] D. Popov (LHCb Collaboration), *EPJ Web Conf.* **214**, 02043. 4 p (2019)

- [11] T. Sjöstrand, S. Mrenna, P. Skands, *Comput. Phys. Commun.* **178**, 852 (2008), [0710.3820](#)
- [12] T. Sjöstrand, S. Mrenna, P. Skands, *JHEP* **05**, 026 (2006), [hep-ph/0603175](#)
- [13] I. Belyaev et al., *J. Phys. Conf. Ser.* **331**, 032047 (2011)
- [14] D.J. Lange, *Nucl. Instrum. Meth.* **A462**, 152 (2001)
- [15] J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Dubois et al. (Geant4 collaboration), *IEEE Trans.Nucl.Sci.* **53**, 270 (2006)
- [16] S. Agostinelli et al. (Geant4 collaboration), *Nucl. Instrum. Meth.* **A506**, 250 (2003)
- [17] M. Clemencic et al., *J. Phys. Conf. Ser.* **331**, 032023 (2011)
- [18] P. Li, E. Rodrigues, S. Stahl, *Tech. Rep. CERN-LHCb-PUB-2021-005*, CERN, Geneva (2021), <https://cds.cern.ch/record/2752971>
- [19] R. Brun, F. Rademakers, *Nucl. Instrum. Meth. A* **389**, 81 (1997)