PROCEEDINGS OF SCIENCE

Gauss and Gaussino: the LHCb simulation software and its new experiment agnostic core framework

Michał Mazurek,[a,*] **Marco Clemencic**[a] **and Gloria Corti**[a]

[a] *CERN,*
Espl. des Particules 1, Meyrin, Switzerland

E-mail: michal.mazurek@cern.ch

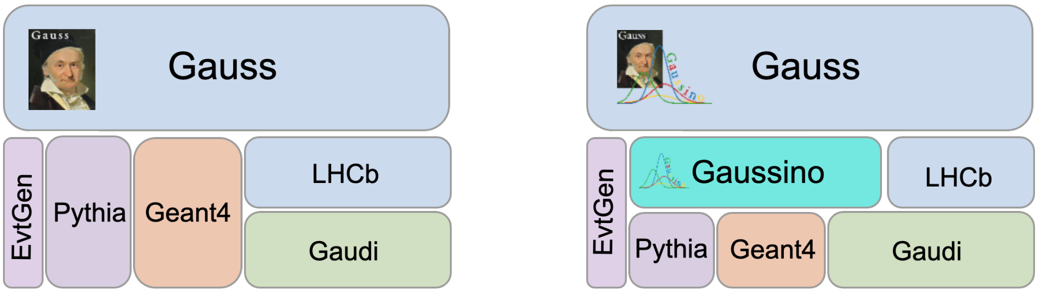
The LHCb experiment is resuming operation in Run3 after a major upgrade. New software exploiting modern technologies for all data processing and in the underlying LHCb core software framework is part of the upgrade. The LHCb simulation framework, Gauss, had to be adapted accordingly, with the additional constraint that it also relies on external simulation libraries. At the same time, a decision was taken to consolidate the simulation software and extract all generic components into a new core experiment-independent framework, called Gaussino. This new core simulation framework allows easier prototyping and testing of new technologies where only the core elements are affected. It relies on Gaudi for general functionalities and the Geant4 toolkit for particle transport, combining their specific multi-threaded approaches. A fast simulation interface to replace the Geant4 physics processes with a palette of fast simulation models for a given sub-detector is the most recent addition. Geometry layouts can be provided through DD4hep or experiment-specific software. A built-in mechanism to define simple volumes at configuration time and ease the development cycle is also available. A plug&play mechanism for modeling collisions and interfacing generators like Pythia8 and EvtGen is provided. We will describe the structure and functionality of Gaussino and how the new version of Gauss exploits Gaussino's infrastructure to provide what required for the simulation(s) of the LHCb experiment.

41st International Conference on High Energy physics - ICHEP2022
6-13 July, 2022
Bologna, Italy

[*] Speaker

1. Introduction

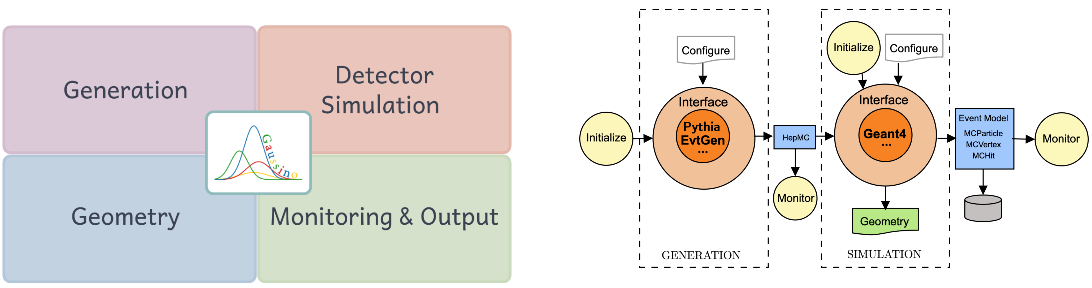
The LHCb experiment [1, 2] is a single-arm spectrometer with a forward angular coverage at the Large Hadron Collider (LHC) at CERN designed for the study of particles containing b or c quarks. The experiment has just resumed data taking after a major upgrade and will operate with higher luminosity and trigger rates than in the previous LHC runs. These high rates are very challenging for the processing of the data and, as a result, major changes in the software applications were introduced as part of the upgrade program for Run3, including the simulation software.



(a) Gauss current dependencies **(b)** Gauss-on-Gaussino dependencies

Figure 1: Dependencies in the simulation software stack before and after upgrade.

The amount of simulated samples for given physics analysis is already limiting the precision of some of the LHCb measurements and will do so more and more in the future. Therefore, the LHCb simulation software Gauss has been redesigned [3] in order to provide support for new software technologies and methods that will enable to fit the statistical requirements of simulated samples for Run3 within the computing resources allocated for the next few years [4, 5]. Focus has been put on the use of multi-threading and fast simulations, including machine learning options. Adaptation to new detector description tools and many other changes introduced in the LHCb common core software are also implemented. The LHCb simulation team decided to use this as an opportunity to consolidate the Gauss software [6–9] and extract all its generic components into a new experiment-independent core simulation framework, called Gaussino, as shown in Figure 1. Gaussino allows easier prototyping and testing of new technologies where only the core elements are affected. The new version of Gauss, referred to as Gauss-on-Gaussino, exploits the Gaussino's infrastructure and provides all the additional functionalities specific to LHCb.



(a) Main building blocks of Gaussino

(b) Execution structure of Gaussino with its two consecutive phases: event generation and detector simulation.

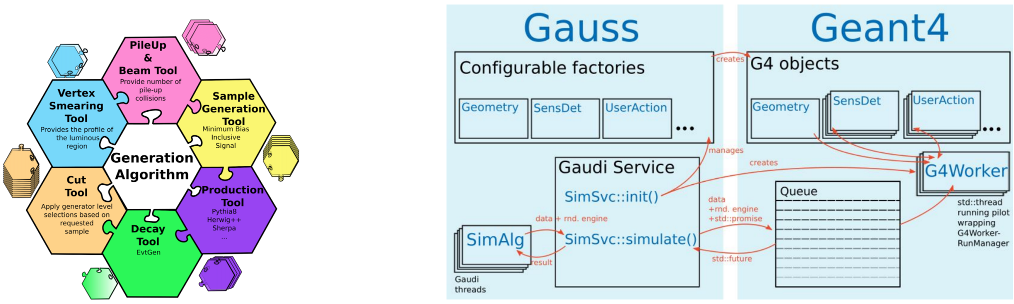
Figure 2: Gaussino inherited from Gauss a similar structure and modularity.

2. Structure of the new framework

Gaussino can be seen as an assembly of modular core components that were already identified and exploited in Gauss. As shown in Figure 2a, these main components are: the generation of events, the detector simulation, geometry services, and monitoring and saving of the output. A dedicated python configuration is available for each of these building blocks. The main execution (Fig. 2b) of Gaussino is structured in two main independent phases - event generation and detector simulation - that can be run together or separately.

The generation phase, strictly following the original design in Gauss [10], is structured as an algorithm (Fig. 3a) that delegates to dedicated tools specific tasks of the generation, e.g. conditions and production of the proton-proton collisions, decay of the particles, etc. Most of the events in Gauss are generated with Pythia8 [11] and particles decayed with EvtGen [12], however, the infrastructure provides a plug&play mechanism, so that other generators can be easily integrated.

The detector simulation phase, in which particles are transported and their physics processes with materials are handled when traversing through physical volumes, is delegated to the Geant4 [13, 14] toolkit. The interface to Geant4, shown in Figure 3b, was completely redesigned in order to provide a thread-safe communication between Gaudi [15–17] and Geant4 different multi-threading environments. HepMC3 [18] is used as an exchange format within the generator phase and between the generator and simulation phases. It guarantees thread-safety and a smaller memory footprint. Although currently Gaussino uses the LHCb event model objects to save its combined generation and simulation output, investigation is in progress to adopt a more generic candidate, e.g. EDM4hep [19].



(a) Modular structure of the generation phase with separate tools that are used for handling various tasks.

(b) Schematic view of the Gaussino's interface to Geant4. It is thread-safe and guarantees communication between threads spawned by Gaudi and Geant4.

Figure 3: Main components of the generation and simulation phase in Gaussino.

Precise and efficient geometry description of the experimental apparatus is crucial when simulating the physics processes during particle transport. Gaussino provides a generic service to support how to provide relevant information to Geant4 from specific geometry descriptions, e.g. DD4hep [20]. In LHCb, DD4hep has been chosen as the new detector description toolkit for Run3 but an additional challenge arises from the need to produce with the new software also samples for Run 1 and Run 2 physics analyses, for which the LHCb custom legacy detector description has to be used. Gauss-on-Gaussino has been designed to extend the generic geometry service to support the multiple detector description tools. Moreover, Gaussino has been equipped with an additional, internal geometry service: ExternalDetector. It is a very generic tool that can be used to run Gaussino in a standalone mode or to allow obtaining additional information for non-standard studies, e.g. production of training data sets for machine learning models, evaluation of new sub-detectors, etc.

3. New features in view of Run 3

One of the major improvements introduced in Gaussino is an inter-event-based parallelism. Therefore, many of its software elements had to be rewritten in order to guarantee a thread-safe execution, as well as a flawless communication with external simulation libraries having their own, custom multi-threaded implementations. A special interface to Geant4 was designed for this purpose and is shown in Figure 3b. It features a dedicated simulation service whose primary role is to ensure that all the necessary information is passed on to and retrieved from Geant4 objects created via Gaudi tools, which act as configurable factories.

The multi-threading memory and throughput performance purely for the generation phase and for the generation and detector simulation phase combined are discussed in Ref. [6, 8]. The improvement is, nevertheless, not sufficient to produce the whole set of simulated samples necessary for Run3. Further optimization of the time spent in the detector simulation, where the modeling of particles' propagation through matter and of the physics processing occurring wherein, is needed.

The LHCb simulation team invested significant effort in recent years to introduce fast simulation techniques in Gauss, many of which are already being used [21, 22], while other are still being developed. A dedicated FastSimulation [23, 24] interface has been introduced in Gaussino to fully exploit the mechanisms available in Geant4 to invoke fast simulations for a given detector. Point library [25] and a model based on generative adversarial networks (GANs) [26], designed to replace the Geant4 detailed description of the physics processes occurring in the LHCb calorimeters, will be the first models to exploit this FastSimulation interface. The Gauss-on-Gaussino framework is also being extended to host the LHCb ultra-fast parametrization, Lamarr [27].

4. Conclusions

Monte Carlo simulations are key to the design and commissioning of new detectors as well as the interpretation of physics measurements. In the last few years, the LHCb simulation software Gauss underwent major rewriting in view of Run3 to introduce new software technologies, such as Gaudi and Geant4 multi-threading, fast simulation techniques including machine-learning-based solutions, new detector descriptions, etc. Gaussino, a new experiment-independent simulation framework that will serve as a core simulation framework for Gauss, was implemented. Gauss-on-Gaussino is the new version of Gauss based on Gaussino and where the LHCb-specific additions are kept. Both Gaussino and Gauss-on-Gaussino are in the final testing phase and the first beta version for production tests is planned by the end of 2022.

References

[1] LHCb Collaboration, *The LHCb detector at the LHC*, *JINST* **3** (2008) S08005.

[2] LHCb Collaboration, *LHCb detector performance*, *Int. J. Mod. Phys.* **A30** (2015) 1530022 LHCB-DP-2014-002, CERN-PH-EP-2014-290, [1412.6352].

[3] LHCb collaboration, *LHCb Upgrade Software and Computing*, 2018. CERN-LHCC-2018-007.

[4] LHCb collaboration, *LHCb CPU Usage Forecast*, 2019. LHCB-FIGURE-2019-018.

[5] LHCb collaboration, *Computing Model of the Upgrade LHCb experiment*, 2018. LHCb-TDR-018.

[6] B.G. Siddi and D. Müller, *Gaussino - a Gaudi-Based Core Simulation Framework*, in *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pp. 1–4, 2019, DOI.

[7] D. Müller, *Adopting new technologies in the LHCb Gauss simulation framework*, *EPJ Web Conf.* **214** (2019) 02004.

[8] LHCb collaboration, *Performance of a multithreaded prototype for the future LHCb simulation framework (Gauss-on-Gaussino)*, Oct, 2019. LHCB-FIGURE-2019-012.

[9] M. Mazurek, G. Corti, M. Kreps, A. Davis, D. Popov and B.G. Siddi, *New software technologies in the LHCb Simulation*, . LHCb-PROC-2022-003.

[10] I. Belyaev et al., *Handling of the generation of primary events in Gauss, the LHCb simulation framework*, *J. Phys. Conf. Ser.* **331** (2011) 032047.

[11] T. Sjöstrand, S. Mrenna and P. Skands, *A brief introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852 [0710.3820].

[12] D.J. Lange, *The EvtGen particle decay simulation package*, *Nucl. Instrum. Meth.* **A462** (2001) 152.

[13] Geant4 collaboration, *Geant4: A simulation toolkit*, *Nucl. Instrum. Meth.* **A506** (2003) 250.

[14] Geant4 collaboration, *Geant4 developments and applications*, *IEEE Trans.Nucl.Sci.* **53** (2006) 270.

[15] G. Barrand, I. Belyaev, P. Binko, M. Cattaneo, R. Chytracek, G. Corti et al., *GAUDI — A software architecture and framework for building HEP data processing applications*, *Computer Physics Communications* **140** (2001) 45. CHEP2000.

[16] M. Clemencic, H. Degaudenzi, P. Mato, S. Binet, W. Lavrijsen, C. Leggett et al., *Recent developments in the LHCb software framework gaudi*, *Journal of Physics: Conference Series* **219** (2010) 042006.

PoS(ICHEP2022)225

[17] C. Bozzi, S. Ponce and S. Roiser, *The core software framework for the LHCb Upgrade*, *EPJ Web Conf.* **214** (2019) 05040.

[18] A. Buckley, P. Ilten, D. Konstantinov, L. Lönnblad, J. Monk, W. Pokorski et al., *The HepMC3 event record library for Monte Carlo event generators*, *Computer Physics Communications* **260** (2021) 107310.

[19] F. Gaede, G. Ganis, B. Hegner, C. Helsens, T. Madlener, A. Sailer et al., *EDM4hep and podio - The event data model of the Key4hep project and its implementation*, *EPJ Web Conf.* **251** (2021) 03026.

[20] M. Frank, F. Gaede, C. Grefe and P. Mato, *DD4hep: A detector description toolkit for high energy physics experiments*, *Journal of Physics: Conference Series* **513** (2014) 022010.

[21] M.P. Whitehead, *A Palette of Fast Simulations in LHCb*, Jul, 2018. LHCb-TALK-2018-302.

[22] D. Müller, M. Clemencic, G. Corti and M. Gersabeck, *ReDecay: a novel approach to speed up the simulation at LHCb*, *The European Physical Journal C* **78** (2018) .

[23] LHCb collaboration, *Performance of the fast simulation interface in Gauss-on-Gaussino*, Sep, 2021. LHCB-FIGURE-2021-004.

[24] M. Mazurek, G. Corti and D. Müller, *New Simulation Software Technologies at the LHCb Experiment at CERN*, *COMPUTING AND INFORMATICS* **40** (2021) 815–832. LHCb-PROC-2021-011.

[25] M. Rama and G. Vitali, *Calorimeter fast simulation based on hit libraries LHCb Gauss framework*, *EPJ Web Conf.* **214** (2019) 02040.

[26] F. Ratnikov, *Generative Adversarial Networks for LHCb Fast Simulation*, Nov, 2019. LHCb-TALK-2019-403.

[27] Lucio Anderlini et al., *Lamarr: the ultra-fast simulation option for the LHCb experiment*, 2022. These proceedings.