How the Monte Carlo production of a wide variety of different samples is centrally handled in the LHCb experiment

G Corti[1], Ph Charpentier[1], M Clemencic[1], J Closier[1], B Couturier[1], M Kreps[2], Z Màthè[1], D O'Hanlon[2], P Robbe[1,3], V Romanovsky[4], F Stagni[1] and A Zhelezov[5]

[1] CERN, Geneva, Switzerland
[2] Department of Physics, University of Warwick, Coventry, United Kingdom
[3] LAL, Université Paris-Sud, CNRS/IN2P3, Orsay, France
[4] Institute for High Energy Physics (IHEP), Protvino, Russia
[5] Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany

E-mail: `gloria.corti@cern.ch`

Abstract. In the LHCb experiment a wide variety of Monte Carlo simulated samples needs to be produced for the experiment's physics program. Monte Carlo productions are handled centrally similarly to all massive processing of data in the experiment. In order to cope with the large set of different types of simulation samples, necessary procedures based on common infrastructures have been set up with a numerical event type identification code used throughout. The various elements in the procedure, from writing a configuration for an event type to deploying them on the production environment, from submitting and processing a request to retrieving the sample produced as well as the conventions established to allow their interplay will be described. The choices made have allowed a high level of automation of Monte Carlo productions that are handled centrally in a transparent way with experts concentrating on their specific tasks. As a result the massive Monte Carlo production of the experiment is efficiently processed on a world-wide distributed system with minimal manpower.

1. Introduction

In the LHCb experiment [1, 2], all massive data processing is handled centrally by a Production Team. This includes the production of the Monte Carlo (MC) samples necessary for the physics analysis which requires the usage of a considerable fraction of the computing resources available to the experiment.

The productions of MC for LHCb Run1 physics analysis has been ongoing since December 2011 with over 9 billions of events produced.

The MC productions reflect the different running conditions of the data they are supposed to reproduce in terms of beam and trigger settings. Similarly the processing of the simulated samples in terms of reconstruction and stripping (i.e. analyses event selection) matches that of the corresponding data. Two main simulation versions have been used with major differences in the simulation software itself (e.g. version of generators [3] and Geant4 [4, 5]).

A wide variety of different types of simulated events has to be produced as each physics analysis needs different sets of signal and background events: over two thousand different types of events have been produced for these well defined Monte Carlo configurations since the end of 2011. It is essential to keep the configurations for the various sets of samples consistent between each other not only for a given physics analysis but also because some of the samples are shared between different analyses. It is also necessary to ensure that any sample produced is available in a transparent way to the whole LHCb collaboration to use the computing resources efficiently and allow reproducibility of the results. For this reason it was decided early on in the collaboration that all MC samples for physics analysis should be produced centrally, leading to the establishment of procedures based on common infrastructures.

In order to cope with the large set of different types of MC events needed by the experiment a numerical *Event Type IDentification* code has been devised and is used throughout the whole procedure, from the automatic generation of the configuration of the relevant simulation software to the customization of the production requests and the identification of the samples produced. A detailed description of the steps and actors in the procedure and their interplay will be given in this paper. The automation put in place has allowed to efficiently process the MC production of the LHCb experiment on a world-wide distributed system with an average turn around of about two weeks for a normally sized standard production of $O(10\,M)$ events with a team of only a few people.

2. Procedure for Monte Carlo productions

The first step when deploying a new Monte Carlo production version is to establish a well defined processing with stable simulation software and online conditions and offline processing corresponding to the data it has to mimic.

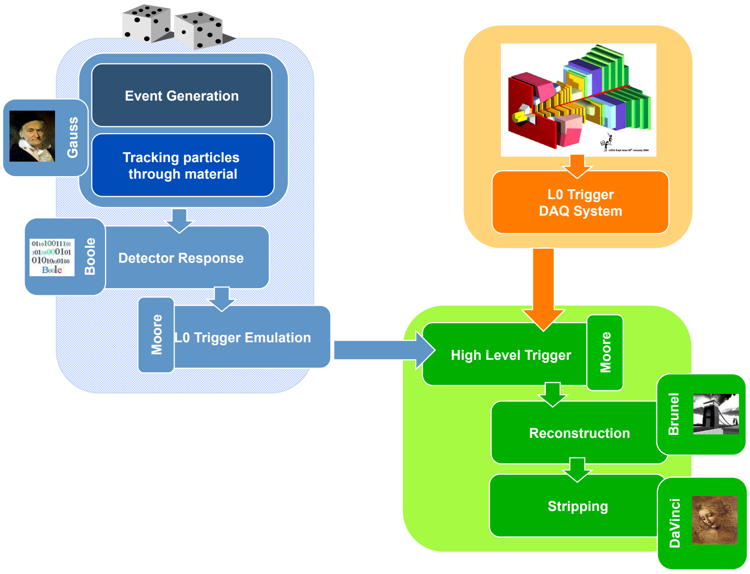

Figure 1. Outline of the LHCb data processing. On the left in blue the processing of Monte Carlo samples, the Data Taking in orange on the right. The Reconstruction and Stripping for both the MC and the data collected by the DAQ of the experiment are in green.

The production of Monte Carlo samples with dedicated simulation software replaces the Data Taking and has to reproduce the same data taking conditions in terms of beam settings and the most representative trigger configuration. The data produced by the simulation software is then processed through the same identical Reconstruction and Stripping (i.e. analysis selections) of the data collected by the Data Acquisition (DAQ) system.

A sketch is shown in figure 1 where each step outlined corresponds to a processing step in the production system of LHCb with the name of the application used indicated on the side: *Gauss* [6] for the event generation and tracking of particles through the detector layout, *Boole* for mimicking the detector response and *Moore* configured to emulate the L0 trigger are simulation specific applications while *Moore* configured for High Level Trigger (HLT), *Brunel* for the reconstruction and *DaVinci* for the stripping are for all data processing. Each step is configured independently and the steps are combined in sequence to define a processing pass. This major endeavor requires non negligible time to set up and commission but once defined, it is frozen for a given set of data taking periods. For LHCb Run1 two major simulation versions have been set up for different data taking and processing configurations, with additional configurations for different generators and some minor version deployed to make available small changes in the simulation software and newer offline re-processing. Nevertheless it is necessary to continuously deploy new types of events as physics analyses are developed and evolve.

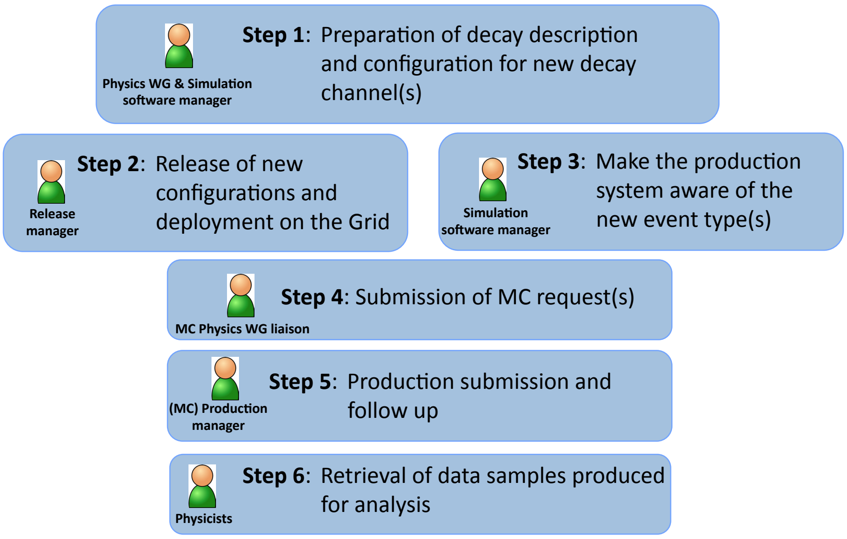
Physics WG & Simulation software manager
Step 1: Preparation of decay description and configuration for new decay channel(s)
Release manager
Step 2: Release of new configurations and deployment on the Grid
Simulation software manager
Step 3: Make the production system aware of the new event type(s)
MC Physics WG liaison
Step 4: Submission of MC request(s)
(MC) Production manager
Step 5: Production submission and follow up
Physicists
Step 6: Retrieval of data samples produced for analysis

Figure 2. Outline of the LHCb procedure for production of new event type samples. Each step in the procedure is described in sequence (step 2 and 3 can be carried out in parallel) and the actors for each step are indicated.

An outline of the various steps and actors in the procedure that has been established in order to obtain a sample of a new event type for a given analysis is shown in figure 2. The first step consists in the preparation of the configuration for new event types, mostly new decay channels of given signal particles. This step is carried out by individual physicists with the help of the Managers of the Simulation Software. Once a set of new configurations is ready they are released by the Simulation Release Manager of the package containing them. The package new release is deployed on the Grid by the LHCb software deployment shifters. In parallel, at release time, the Simulation Release Manager makes the production system aware of the new event types registering them in the LHCb Bookkeeping system. When both of these steps are completed, MC production requests can be submitted by the Physics Working Group (PhWG) MC liaisons who are appropriately trained and have collected the needs of their Working Group (WG). Once a request is submitted the MC Production Manager receives an automatic notification and after final verifications submits the actual production and follows it up. Finally any physicist in the LHCb collaboration can retrieve the samples produced via the LHCb Bookkeeping system.

2.1. Event types configuration

In LHCb the majority of MC samples produced are proton-proton interactions at LHC collision energies with specific decays of b or c hadrons, although other types of events are also made. The main generator used for the modeling of the proton-proton collisions is Pythia [9] while EvtGen [10, 11] is used to model the decay of particles. EvtGen has been originally developed for BaBar and CLEO to provide detailed decay models for the decay of B and D mesons and has been later extended to work in a proton-proton environment and provide decays for all particles. It is currently maintained by LHCb. The default behavior of the various decays is governed by a general *DECAY.DEC* table where all know decay modes for all particles are listed. User decay files are used to force a given decay for a *signal* particle via a specific decay model. In LHCb an extended version of the user decay files has been set up with a steering section to generate the specific configuration of the generators to be used at run time to produce the given sample. The steering section of a typical user decay file may look as follows:

```
# EventType:  11140005
#
# Descriptor: {[[B0]nos -> mu+ mu- (K*(892)0 -> K+ pi-)]cc,
[[B0]os -> mu- mu+ (K*(892)~0 -> K- pi+)]cc}
#
# Nickname:   Bd_Kstmumu,phsp=DecProdCur,MomCut
#
# Cuts:       DaughtersInLHCbAndWithMinP
#
# Documentation: Decay products in acceptance and minimum momentum cut
# EndDocumentation
#
# PhysicsWG:   RD
# Tested:      Yes
# Responsible: John Doe
# Email:       John.Doe@mail.address
# Date:        20110928
#
```

The first keyword in the steering section is the **Event Type ID**, an eight digits number to uniquely identify each decay file, associated options and sample produced. Conventions have been established for the meaning of each of the eight digits based on the nature and topology of the decay. A script makes use of the value of the digits in `Event Type ID` to automatically produce the configuration of the generator. Additional configurations are steered by other keywords as is the case for example of `Cuts` that correspond to given C++ classes to be used to select only a subset of events. A short mnemonic, a `NickName` is used to provide a unique human readable identifier of the decay file and matches its name. A descriptor of the decay in a form usable by LHCb analysis software is also given. Finally some explanation and provenance of the decay file are described via a set of keywords (`Documentation, PhysicsWG, Tested,` `Responsible, Email, Date`) that are used in the automatic publication on the web of the documentation of the release.

A dedicated package called *DecFiles*, contains all event types and their configurations files automatically generated by a script either on demand or at release time.

2.2. Release and deployment of event types in production

Decay files are continuously made by physicists in the collaboration and added to the SVN code repository of the *DecFiles* package. The managers of the package provide help in the writing of the decay files and verify the compliance to the rules with a set of tools developed for the task. The tag collector of the experiment software is used to notify of the need to release a new given event type. Automatic tests in the nightly build system are run for all new event types committed to verify they can be processed in the production environment. The release of the *DecFiles* package is independent and asynchronous from that of the simulation application that loads it at run time: major version numbers are used to ensure compatibility between the two. The deployment of the *DecFiles* package on the distributed production system used by the experiment is handled centrally via common LHCb distribution tools [12]. Tar-balls of the package are made in the release build system and installed in the LHCb CVMFS [13] repository for distribution to the LHCb production sites. Once that is done the version of the newly released *DecFiles* to be used is specified in the production system.

A dedicated step in the release procedure is to register the list of the newly deployed event types in the LHCb Bookkeeping database [7, 8]. that makes them available to the production system. The LHCb Bookkeeping system is a metadata management system that stores the conditions relative to jobs, files and their metadata as well as their provenance information in an organized way. It is based on Oracle RDBMS with the data presented to the user as a hierarchical structure in a tree-like format.

2.3. Use of Event Types in the production system

The production system provides a Web User Interface with separate pages dedicated to various tasks accessible to people with specified roles.

Simulation conditions in terms of beam energy, detector geometry, generator used, etc. are described via a dedicate `Simulation Conditions` page. A `Request Manager` page allows to link together the steps with all applications to be executed and to define it as a model for production requests. Simulation conditions and processing through all applications are fixed when launching a new Monte Carlo production configuration while the *Event Type ID* is left as a free parameter.

In the *Gauss* application step the version of the *DecFiles* package is listed explicitly. A `Step Manager` interface allows to clone an existing step and modify it without having to remake it from scratch. When a new *DecFiles* release is made available the existing *Gauss* steps are cloned and the version of *DecFiles* updated without changing the rest of the configuration. Requests, of which models are a specialization, can be modified before submission to replace a single step. This feature is used for the final action to replace an older *Gauss* step with one cloned from it and using a newly released *DecFiles* version. The feature is also used in the MC filtering productions where only events passing a specific final stripping selection are kept: for them different configurations from specific PhWG packages must be used and are in some cases unique to a single production request.

The MC liaisons use the feature provided by the `Request Manager` interface to create a simulation request by cloning a model, selected from the list of those available. They then complete the request specification based of the needs of their PhWG selecting the *Event Type ID* from the list of those available and specifying the number of events to be produced at the end of the processing.

2.4. Production of MC productions in the LHCbDIRAC system

The production request system is part of the LHCbDIRAC production system [14] that is used to schedule the thousands of jobs for a given request, execute and monitor them on the available computing resources.

From the LHCbDIRAC production system viewpoint all that has been described so far can be summarized as follow:

1. The Simulation Application Managers generate the application steps, i.e. a job step description
2. The Simulation Application Managers link steps together creating production requests
3. The Physics Working Group via their MC liaison make requests for given event types by cloning a model and selecting the event type from the list of those available for production

At this point the control passes to the LHCb Production team and

4. The MC Production Manager submits the productions.

The LHCbDIRAC production system has a set of dedicated production templates specialized for MC productions. Once the productions are submitted the requests descriptions become workflows and then DIRAC [15] jobs that are executed on the available computing resources. MC productions are in fact split into n sub-productions each with a set of subsequent steps, for example to have one sub-production running all simulation applications in sequence and one sub-production running all offline applications afterward. The *Gauss* simulation application is highly CPU time consuming and only a very small number of events, of the order of few hundreds, can reliably be produced per job. On the other hand the trigger and reconstruction applications are several orders of magnitude faster and can process many more events in a single job. In addition the size of a reconstructed event is small (DST) and only a limited amount of the MC truth data is included in a MC-DST: this would lead to many very small files with a resulting limitation on data access. The subdivision of a production request in n sub-productions allows for example to optimize the number of input samples for MC reconstruction sub-productions coming from many jobs of the MC simulation sub-production.

Initially the number of jobs in the productions was determined by the MC Production Manager after some dedicated small tests. In the last year *elastic Grid jobs* [16] have been introduced in LHCbDIRAC and productions are now extended and closed automatically once the requested number of events has been produced.

2.5. MC requests status and priority

The requests system keeps track of the status of a request and ensure that the next transition can take place with e-mail notifications sent automatically by the system to relevant people. The request status at each stage in the cycle and the roles defined in the system that are allowed to change that status are outlined in figure 3.

The appropriateness of a request for the physics program of the experiment in terms of priority (relevance and urgency) and amount of data to be produced is under the control of the Physics management.

The priority of the running jobs is determined by the MC Production manager based on the computing resources available taking into account not only all accepted MC production requests but also the processing activities for real data.

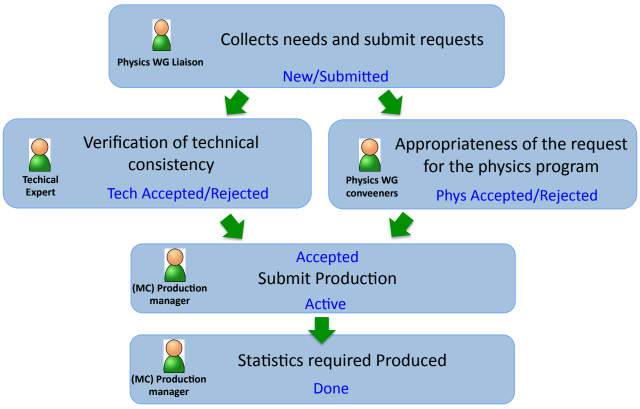

Figure 3. Outline of the workflow of MC production requests. The people performing the changes are listed with the request status in blue.

3. Retrieval of MC datasets

The data produced are automatically registered by the production system into the Bookkeeping database where they can be found in folders specifying the event type, simulation conditions and processing chain. The metadata information of the tasks executed on the Grid to produce an MC dataset are uploaded at the end of the job that produced them as provenance of the data. Any physicist in the LHCb collaboration with a LHCb Virtual Organization certificate can access the Bookkeeping to find and read any MC dataset centrally produced. The data is presented in a tree-like structure and can be searched first via the `Event Type ID` or the `Simulation Condition` of the production. All details relative to an MC dataset are available via its Simulation Condition and Processing Pass.

Conventions have been established for the format of the *Simulation Conditions* as to uniquely describe all possible Beam and Detector configurations the samples are produced with.

The *Processing Pass* is also available in the tree structure presented to the user by the Bookkeeping where each folder indicates a processing step that has been set as visible. Not all steps are set as visible to avoid unnecessary folders, for example only the *Gauss* simulation step is set as visible while the *Boole* step is set an invisible since their combination is considered as a unique `Simulation Version`. For an MC dataset for physics analysis this is normally followed by the `Trigger Configuration Key` used in the L0&HLT and that corresponds to that of the data taking the simulation is representing and by the `Reconstruction` and `Stripping Versions` of the (re-)processing of the data the simulation has to match.

4. Conclusions

Conventions and procedures have been implemented in LHCb to handle Monte Carlo productions centrally in a transparent way. LHCb standard tools and common computing infrastructures are used throughout. A unique numerical identifier for the MC samples, *Event Type ID*, is a key element and is used in all the steps of the software deployment and productions. The choices made have allowed to automate and trace all steps of MC productions. Experts can concentrate on their specific tasks: physicist and simulation software experts on the configuration of the application for each different type of events, the production team on job submission, monitoring and data storage. As a result the massive MC productions of the LHCb experiment are carried out in a transparent and efficient way on a world-wide distributed system by a team of one MC Production Manager (part-time) and few Simulation Software Managers (minimal-time).

References

[1] Alves A Augusto J *et al* (LHCb) 2008 *JINST* **3** S08005

[2] Aaij R *et al* (LHCb) 2015 *Int. J. Mod. Phys.* **A30** 1530022 (Preprint 1412.6352)

[3] Belyaev I *et al* 2011 *J. Phys.: Conf. Ser.* **331** 032047

[4] Allison J *et al* (Geant4 collaboration) 2006 *IEEE Trans. Nucl. Sci.* **53** 270

[5] Agostinelli S *et al* (Geant4 collaboration) 2003 *Nucl. Instrum. Meth.* **A506** 250

[6] Clemencic M *et al.* 2011 *J. Phys.: Conf. Ser.* **331** 032023.

[7] Baud J P, Charpentier Ph, Ciba K, Graciani R, Lanciotti E, Màthè Z, Remenska D and Santana R 2012 *J. Phys.: Conf. Ser.* **396** 032023

[8] Stagni F and Charpentier Ph 2012 *J. Phys.: Conf. Ser.* **368** 012010

[9] S̈jostrand T, Mrenna S and Skands P 2008 *Comput. Phys. Commun* **178** 852

[10] Lange D J 2001 *Nucl. Instrum. Meth.* **A462** 152

[11] http://evtgen.warwick.ac.uk/docs/

[12] Clemencic M and Couturier B 2015 LHCb Build and Deployment Infrastructure for run 2 *CHEP2015 Proceedings*

[13] Blomer J, Aguado-Aánchez C, Buncic P and Harutyunyan A 2011 *J. Phys.: Conf. Ser.* **331** 042003

[14] Casajus A *et al* 2012 *J. Phys.: Conf. Ser* **396** 032107

[15] Tsaregorodtsev A *et al* (Dirac Project) 2014 *J. Phys.: Conf. Ser* **513** 032096

[16] Stagni F and Charpentier Ph 2015 Jobs masonry in LHCb with elastic Grid Jobs *CHEP2015 Proceedings*