Ensuring Simulation Quality in the LHCb experiment

26th International Conference on Computing in High Energy & Nuclear Physics (CHEP2023)

Benjamin Couturier, on behalf of the LHCb Simulation & Computing Projects

CERN

9 May 2023

Introduction

Simulation is a key tool in HEP

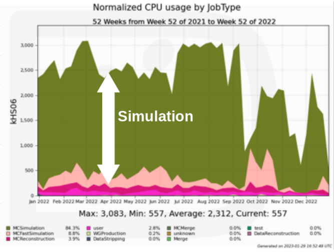
Accuracy & reliability are critical for quality physics results

⇒ **suite of verifications to ensure quality**

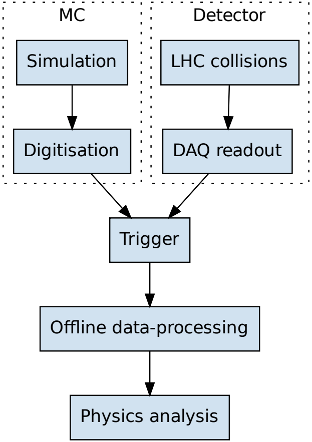
Primary user of computing resources

- More than 95% of offline CPU use in LHCb during 2022

⇒ **important to catch issues early**



LHCb Offline CPU use by production type in 2022



LHCb data-processing chain

Gauss

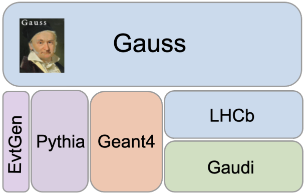
- **Generates** the initial particles and **simulates** their transport through the detector

Gauss-on-Gaussino

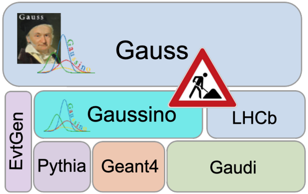
- Incorporates the reimplemented or modernised core features of Gauss
  - ⚠ will replace Gauss in the future [Talk by M. Mazurek]

Boole

- Reproduces subdetectors signals (**digitisation**)
  - *i.e.* MC Hits $\rightarrow$ DAQ RAW data buffer



Gauss structure in Runs 1 and 2



Gauss-on-Gaussino structure in Run 3

Development lifecycle for simulation software

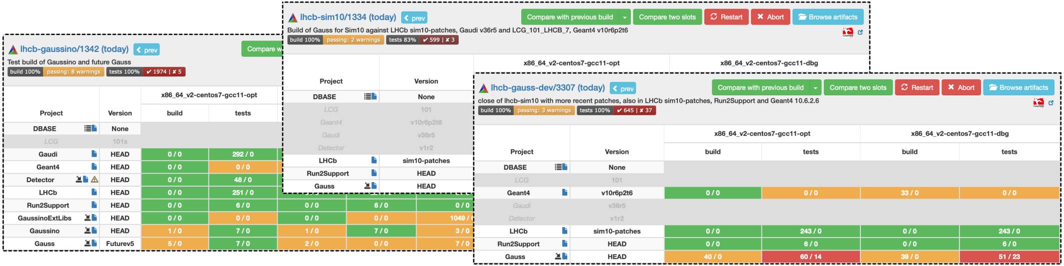
Variations of simulation software stacks

- Multiple versions of the LHCb software stack configured with different tools/versions
- Includes current versions used in production, new versions under commissioning and future versions used as test-beds for new technologies, detector design etc

Nightly builds

- Handled by Jenkins based LHCb build system through various nightly “slots”
- To validate different aspects of the built software stacks, such as:
  - New versions of MC generators and their tuning
  - Upgrade of the underlying software framework (Gaudi, GEANT4)
- Problems occurring in the build process will be reported to the developers

Development lifecycle for simulation software



Nightly builds slots for LHCb simulation software stacks

Initial checks: CI and nightly tests

Simple and quick checks of code quality and functionality

Continuous Integration

- Verify submitted changes to the code
  - Formatting compliance
  - Static code analysis
- Build and deploy documentation
- Accepted changes are applied to nightly builds

Fast feedback directly in GitLab

Nightly tests

- Verify that the built software works
  - Application starts, runs and finalises successfully
  - Correct libraries are picked up
  - Run for every build

Feedback in the LHCb nightly builds interface

✓ Simple and fast to execute without validating the full spectrum of variables

✗ Not enough CPU available to validate the physics

Detailed analysis: PR tests

Comprehensive validation of technical and physics aspects

Performance and regression testing

- Handled by the LHCbPR system
- Simulating $\mathcal{O}(10^3)$ events for physics analysis, thus more time to execute
- Test frequency adapted according to available test resources

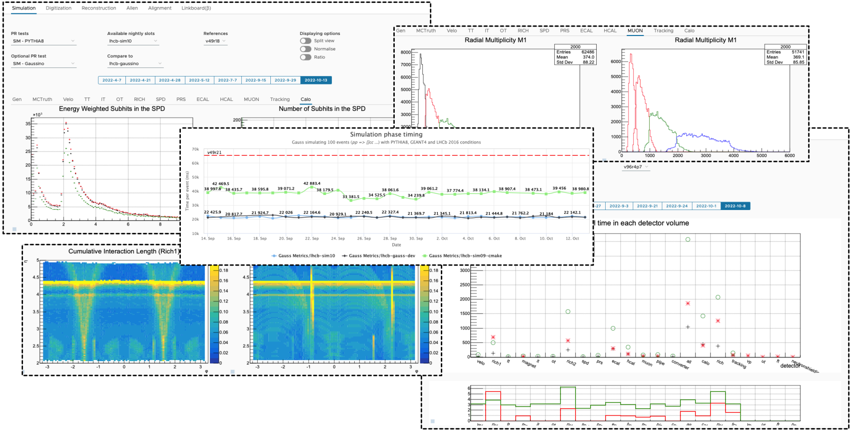
Feedback in the dedicated LHCbPR web application

LHCbPR allows comparing quality measures between different variations of the applications

Capability to support different configurations

GEANT4 PR tests	Gauss PR test
Hadronic cross-section	Gauss simulation validation
Sampling calorimeter	Radiation length and absoption map
Multiple scattering in thin layer	Muon multiple scattering
Simplified RICH simulation	Detailed timing in detector volumes
Gamma to di-lepton conversion	CPU and memory consumption

Detailed analysis: PR tests



Result of PR test, shown on LHCbPR webpage

Simulation data quality

Simulation productions typically generate $\mathcal{O}(10^6)$ events

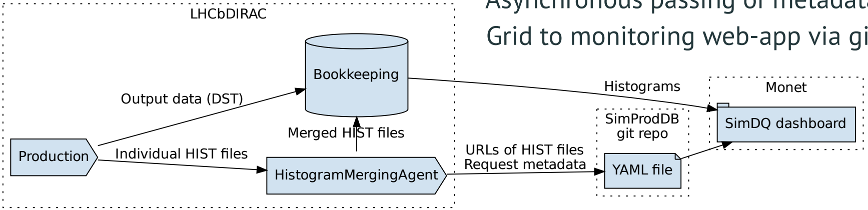
- Previously only relied on whether test jobs succeeded or failed
  - ✗ Data-quality issues typically not spotted until use in analysis

Check histograms of various quantities early on in a production

- Before launching each production request, **test jobs** are run in debug mode
  - Sample of $\mathcal{O}(10^4)$ events produced
  - **Histograms** written to separate files
- New: **Show histograms to shifters** in DQ monitoring web-app (Monet)
  - Also used for online and offline DQ
  - ✓ Allows problems to be **spotted early**
  - Work in progress to add automatic histogram analysis

For more on LHCb simulation production requests: [Talk by C. Burr]

Data quality monitoring for a simulation production request



Asynchronous passing of metadata from Grid to monitoring web-app via git repo

The Grid side (LHCbDIRAC)

- Test production in debug mode
- HIST files merged and uploaded
- Metadata pushed to git repository

Intermediate tasks (git repository)

- CI jobs update timestamps & references
- Webhooks trigger `git pull` in Monet

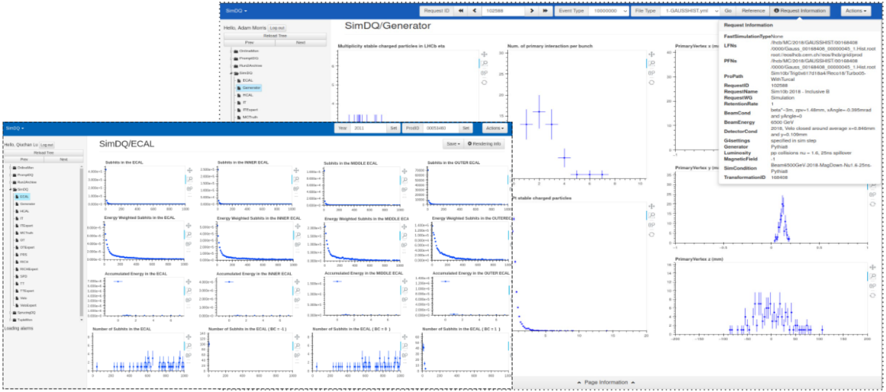
The web monitoring side (Monet)

- Read metadata from git repository
- Browse recent production requests
- Open HIST files and render plots

✓ Same infrastructure as Offline DQ

✓ Uses GitLab's automation features

What the shifters see



Example of Monet interface, in SimDQ mode

Conclusion

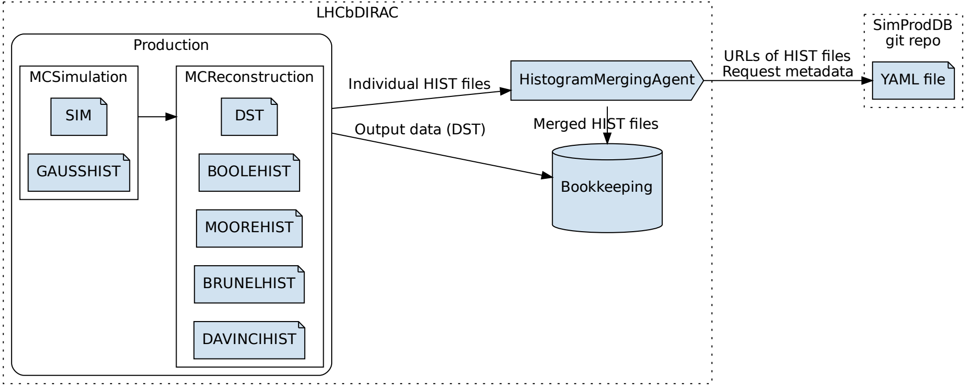
A full suite of verifications is in place for the LHCb Simulation software to ensure the simulation data quality.

- **CI test and nightly test** is developed to verify the integrity of the software with short jobs run every night.
- **PR test** is carried out to verify physics observables in plots and can be compared to references in LHCbPR.
- **SimDQ** is performed to automatically verify the quality of LHCb Monte-Carlo productions in Monet.
  - Implemented simulation DQ in Monet
  - Working on the implementation of the DQ for subsequent steps in the chain

LHCb DQCS shifter helps to use such infrastructures to verify simulation data quality, and alert experts of anomalies and unexpected changes.

Appendix

SimDQ in Monet



SimDQ in Monet