

1.1 LHCb (LARGE HADRON COLLIDER BEAUTY) APPLICATION

The main goal of this application, among other goals, is to test and validate the BalticGrid infrastructure. The means for such test are processes related to CERN experiments. Tests aim to show, how BG operational infrastructure corresponds to the EGEE one. Namely, the efforts in the project were taken to generate processes of high energy physics, to simulate detectors' response and to perform the re-construction task. These efforts are followed by the additional procedures: to store results in form of AOD, to install the LHCb or similar software components and to test BG infrastructure in real data production and data analysis tasks.

The application implemented is to simulate large number of events using the full simulation program, from LHCb experiment of CERN. A single event simulated contains the information about products of proton-proton collision. It is presented in the form which is expected to come from the real detector. Then the data are used to research experimental aspects, in a full scale. The main aspects of such study are:

- to analyze the detector's performance,
- to optimize the detector's design,
- to develop algorithms capable to run on the on-line farm, during real data taking (Trigger algorithms),
- to prepare off-line event reconstruction and analysis.

1.1.1 Requirements and computing procedure

A single LHCb job produces an output file typically corresponding to 500 events of proton-proton collision at the LHC nominal energy. The application requires relatively modern hardware: computers must be equipped with CPUs faster than 1.5 GHz and having at least 1 GB of RAM memory. The main platform for software development has to include i386 compatible processors and Linux operating system. Linux flavour, currently supported is SLC3. Another platform corresponds to MS Windows with MS Visual C++ compiler. Also LHCb VO has to be supported by each site running this application.

The validation of the EGEE and BalticGrid infrastructure is restricted to SLC3 systems only.

Production job consists of a few steps executed on a given working node (WN). Each output of a given step is used as an input for the next step and is processed subsequently. Most files from intermediate steps are deleted. The final output file is usually transferred to the Tier-1 center (by default now is CERN) and stored on a tape.

1.1.2 Prerequisites and software

Jobs are submitted centrally to an execution site, employing LHCb production managers. The execution site has to pass standard LCG tests as well as specific LHCb tests. Dedicated jobs, sent to the site by the LHCb production manager, are responsible for the LHCb software installation. The site is registered in a list of active sites known to the LHCb resource broker.

The main part of application is based on the code developed by the LHCb Collaboration. The general framework is called GAUDI. It incorporates several HEP packages that are used at various stages of data processing.

Application consists of two main modules. The first one, called GAUSS, involves three main steps:

- physics' event generation at the level of particle four-momenta (PYTHIA+EVTGEN),

- transport of particles throughout the detector material (GEANT4),
- generation of an response of electronics, and building the raw data structure of the event (Raw Buffer).

Output file from a single job contains typically about 500 Raw Buffers, one Raw Buffer per event. The second module, called BRUNEL, reads Raw Buffers and performs event reconstruction in various detector systems, e.g. in tracking, in particle identification, in calorimeters or in muon system. The records of output file in ESD format are recorded onto disk and then transferred to the high capacity data center, where they are stored on tapes.

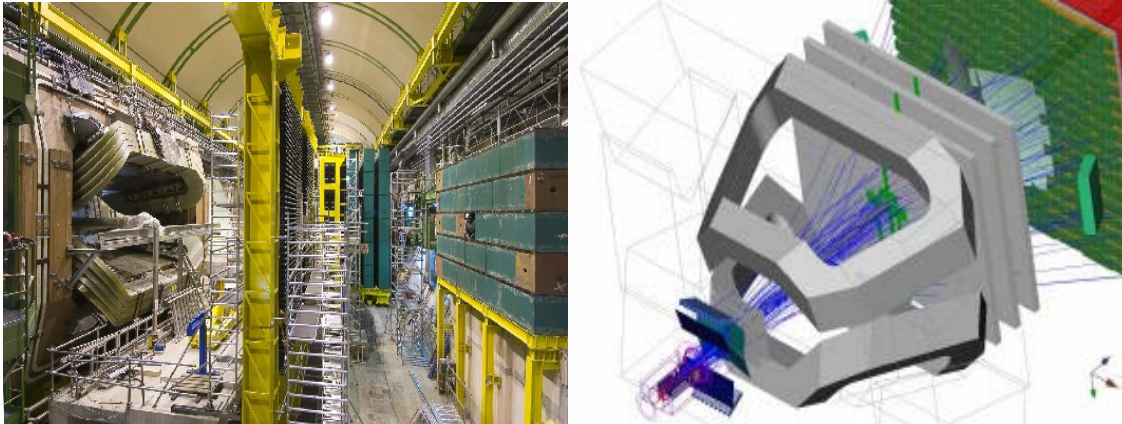


Fig. 8. LHCb detector under the construction (left) and its simulated counterpart (right). On the right picture the blue lines correspond to products of proton-proton collision.

1.1.3 Statistics

MC production corresponding to LHCb jobs has been running on the LCG infrastructure all over the world for several years. In particular, about 2% of the MC production was performed in Cracow and Warsaw during 2006. Daily average is about 40 jobs running in Tier-2 in Cracow.

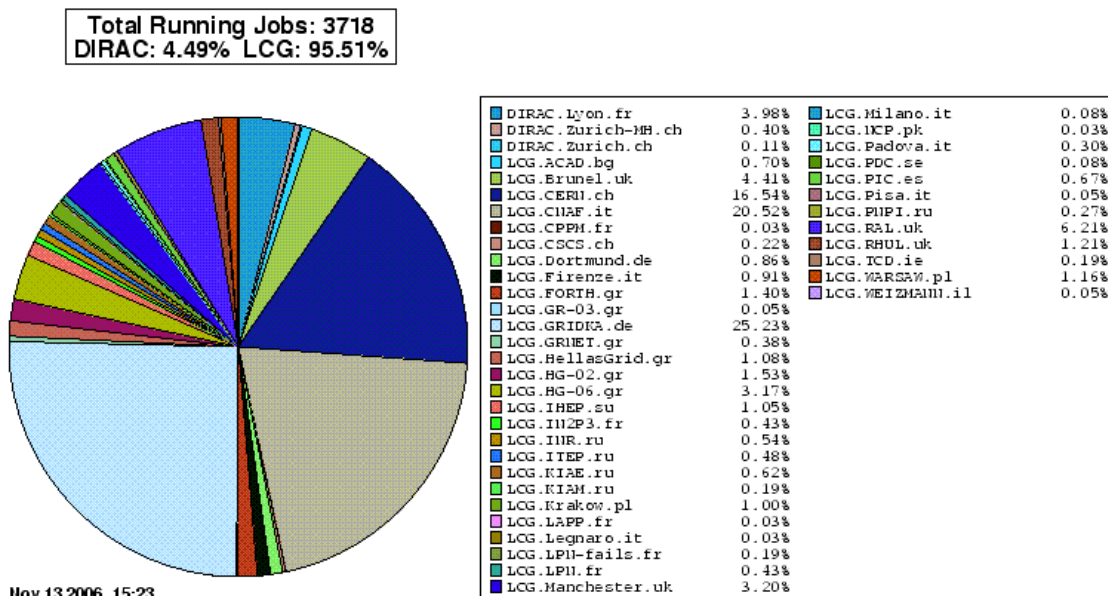


Fig. 9. LHCb monitoring system and snapshot of running jobs for sites participating in the MC production.

Three BalticGrid sites in Lithuania, Latvia and Estonia are to be prepared to use the LHCb MC production for tests of their EGEE functionality. The sites are currently in a preparation stage to support the LHCb VO. After completion they will be registered into the list of the LHCb production sites. Results of the tests will be reported in the next reporting period.

1.1.4 Statistical data analysis using Monte Carlo methods

The principle purpose of this application is to provide a framework for solving statistical problems, employing CPU intensive Toy Monte Carlo methods. The application is being primarily developed in order to study both sensitivity and systematic biases of the CP violation measurements. It can be used to solve other statistical problems as well.

The CP, which transforms matter into anti-matter particle, stands for combination of C – charge conjugation, and P – parity. The violation of CP symmetry implies that the behaviour of matter and anti-matter is different. It is one of the three conditions necessary to explain why the visible Universe is overwhelmingly made of matter.

The CP measurements come as a result of complicated procedure. *B* mesons are produced in hadronic environment of proton-proton collisions. The *B* meson production is over 100 times smaller than the normal one. Moreover, *B* meson decays that are interesting for CP measurements, are relatively rare ranging from 10^{-4} down to 10^{-9} of all *B* decays. Extraction of tiny signal out of the huge background requires sophisticated algorithms, operating already at the level of on-line data taking (reduction from 40 million down to 200 events per second). The data are then reconstructed off-line and CP-violation phenomena are studied, for more than 50 different *B* meson decay modes. In the final step, essential physics parameters are determined by applying the fitting procedure to the data.

1.1.5 Importance and Requirements

The application is used by physicists from the IFJ PAN LHCb group and concerns the analyses of CP violation in B meson decays. The framework of the application is generic and may be exploited in other scientific fields covered by the NA3 task of the BalticGrid.

Full functionality of the application is provided by the ROOT package. The ROOT system provides a set of object oriented frameworks with all the functionality needed to handle and analyse large amounts of data in a very efficient way (see fig. 8). Having the data defined as a set of objects, specialised storage methods are used to get direct access to the separate attributes of the selected objects, without having to touch the bulk of the data. Included are histogramming methods in 1, 2 and 3 dimensions, curve fitting, function evaluation, minimisation, graphics and visualization classes to allow the easy setup of an analysis system that can query and process the data interactively or in batch mode.

Because of the built-in CINT C++ interpreter the command language, the scripting, or macro, language and the programming language are all in C++. The interpreter allows the fast prototyping of the macros since it removes the time consuming compile/link cycle. It also provides a good environment to learn C++. If more performance is needed the interactively developed macros can be compiled using a C++ compiler.

The system has been designed in such a way that it can query its databases in parallel on MPP machines or on clusters of workstations or high-end PC's. ROOT is an open system that can be dynamically extended by linking external libraries. This makes ROOT a premier platform suitable to build data acquisition, simulation and data analysis systems. The ROOT package distribution is

available on most of popular platforms: Linux, UNIX and Windows. While running on the BalticGrid clusters, the Linux version is used.

1.1.6 Computing procedure and software

A Toy Monte Carlo approach is commonly used in HEP experiments. HEP detectors are very complex apparatus, moreover, investigated phenomena are very subtle. The required precise understanding of both the whole chain of data acquisition as well as the analysis is provided by the full simulation program. The full simulation is a very CPU intensive task, thus only limited statistics MC samples can be produced, insufficient to study measurements precisions. Toy Monte Carlo technique is commonly used in such cases. The main idea is to prepare a simplified model of the measurement employing parton distribution functions (PDF), acceptance functions etc., that are derived from the full simulation program, to submit a large number of jobs which execute the procedure with different initial parameters on a GRID and to analyse distributions of outcomes. The final result depends on many other parameters, some of them determine the PDF or a variable essential for final results extraction.

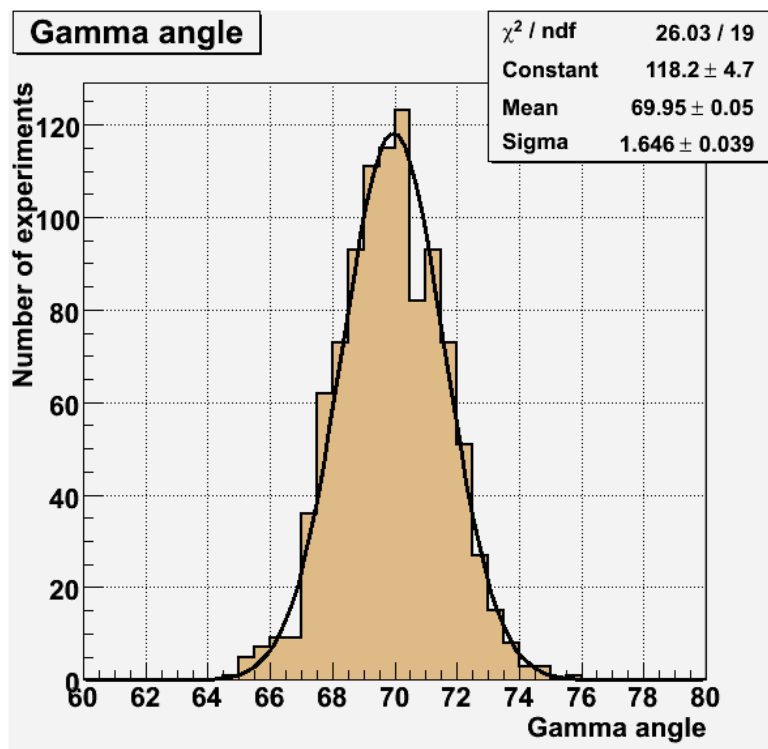


Fig. 10. The typical distribution of results from a bunch of 1000 jobs. The width of the distribution is an estimation of the measurement uncertainty.

Dedicated package which provides tools for building Toy MC program is called RooFit. It was developed for the BaBar experiment and is a part of the ROOT framework. RooFit contains many advanced utilities like high quality pseudo-random number generators, a large set of probability density functions, including conditional probability densities, etc.

The structure of application consists of three main parts:

- **framework part** – based on ROOT, it is responsible for providing the site with ROOT environment. If no installation is found, the ROOT sources are imported and compiled. This is done once, when first job arrives to a given cluster. Thus, the application can be executed on any Linux flavor and some Unix platforms

- **user part** – in the form of ROOT script (C++), although the user script is specific to HEP topics, it can be easily adopted to solve other problems of BalticGrid users
- **grid part** – responsible for submitting jobs and retrieval of results, currently implemented in the form of a set of shell scripts that prepare and submit series of jobs followed by collecting the outputs and final analysis. It is foreseen to integrate the application into Migrating Desktop. The execution of few thousands of jobs is needed in the full production mode to solve one problem. A single job may require several hours of CPU time, depending on complexity level of the problem. The production procedure may require several repetitions.

User script can be executed in two modes:

- interactive mode (C++ interpreter) during the development phase on private workstation
- batch mode for production on the GRID employing compiled version of user code, that results in a faster execution than for an interpreted code.

Statistics. A few thousands of jobs have been executed on BalticGrid sites in total. Initially, several pilot mini productions has been submitted in bunches of a hundred jobs to verify the ROOT installation and correctness of main steering script. Then the simple RooFit scripts were attached and executed, and the results from different sites have been compared. Finally, a simple model of the CP violation measurement has been implemented and extensive productions have been launched. Preliminary physics results have been obtained.