PEFORMANCE MONITORING OF GAUSS, THE SIMULATION SOFTWARE IN LHCb

Fenompanirina ANDRIANALA

HEP-MAD INSTITUTE

UNIVERSITY OF ANTANANARIVO

MADAGASCAR

Abstract

Gauss is the event generation and detector simulation of the LHCb. To simulate the detector, Gauss uses Geant4 which gives information for each particle while it traverses it: the type, the momentum, the originating process…Running a simulation job in Gauss is time consuming because of the time that Geant4 takes to track particle through the detector. This amount of CPU time can be optimized by putting appropriate and reasonable cuts that do not compromise the results.

I- Introduction:

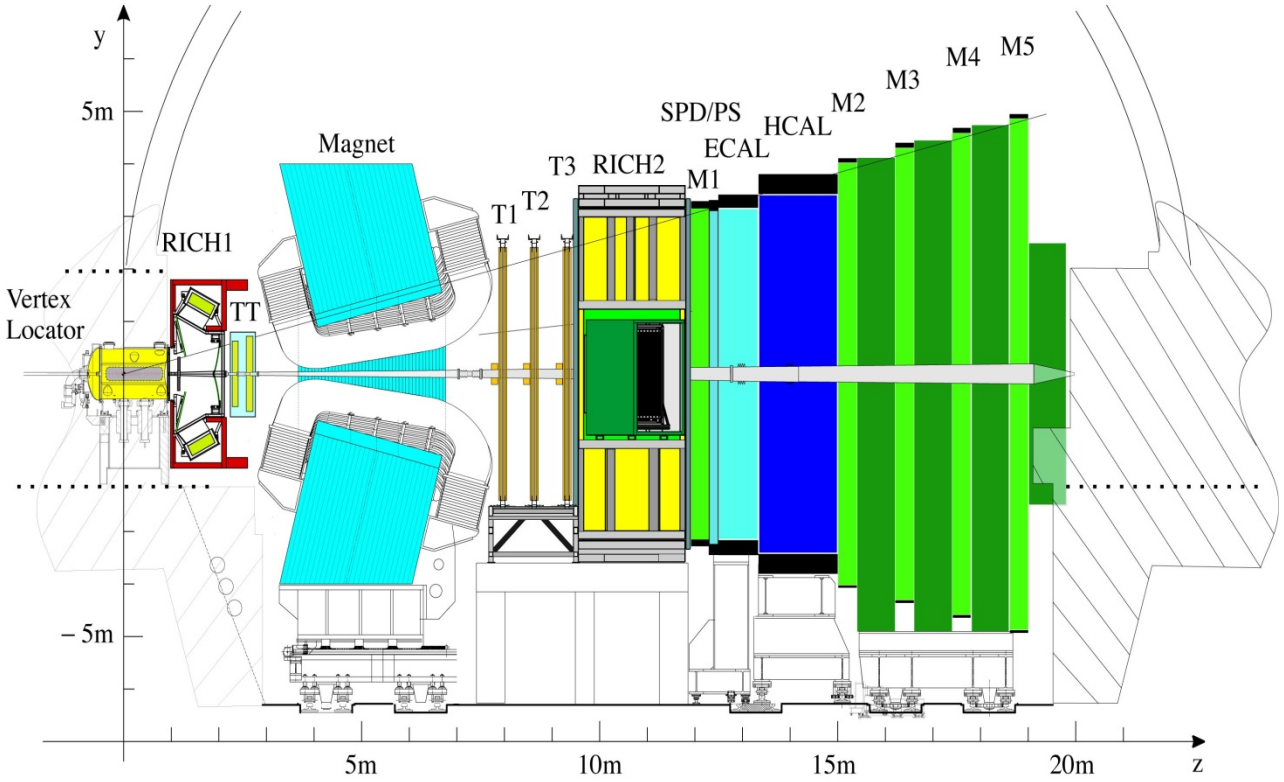
In this report we will have a look at the monitoring of Gauss, the simulation software used in LHCb experiment. Running a job in Gauss is slow due to the fact that it takes a lot of time to track particle through the detector. Gauss uses a toolkit called Geant4 to simulate LHCb detector. We will study the effect on performance in term of CPU time consumption of various choices available for Electromagnetic physic processes and their tuning (production cut) as well as the thresholds (tracking cut) to follow particles through the detector that have been introduced and are specific to Gauss. First of all it will be useful to remind ourselves about the main detector components in order to get an idea of the detection system. Then, the Gauss application will be introduced, followed by its monitoring in which some existed and some was introduced. Finally the results will be discussed and compared in order to see if we should be able to reduce this CPU time at its optimal level.

II- The LHCb detector:

The LHCb experiment is one of the LHC experiments. It is dedicated to study CP violation and other phenomena in B meson decays with very high resolution. This experiment was improved in 1998 and first data are expected on summer 2008.

The main part of the detector can be seen in the figure 1. From left to right, there are the Vertex Locator (VELO), the first Ring Imaging Cherenkov (RICH1) system, the magnet, the tracking system, the second Ring Imaging Cherenkov (RICH2), the calorimeter and the muon detector.

The Vertex Locator is used to measure tracks form primary and secondary vertices around the region of collision. The RICH detector's aim is to distinguish the different charged particles, in order to separate the different b decay channels. The magnet provides very local fields which are useful to find the momentum of charged particles. The trackers are used for reconstruction of particles tracks, giving precise information on coordinates and momentum of charged particles. Calorimeters and muon detector are there to identify the electron, the hadrons and muons.



<u>Figure1</u>: the LHCb detector

III- <u>Gauss application</u>:

1. <u>Definition</u>:

Gauss is the application developed in LHCb for event generation and detector simulation. For the detector simulation it is based on GEANT4. Using Gauss, one can get:

- Monte Carlo Particles: what type, where created, energy and direction
- Monte Carlo Vertices: initial proton-proton interaction, all subsequent decays and interactions with coordinates and their type
- Monte Carlo Hits when each MCParticle crosses a sensitive detector
- Connection between MCParticles and MCVertices and vice-versa

So, Gauss is responsible for tracking particle through the detector.

2. <u>Structure of Gauss application:</u>

Gauss predicts what will happen in LHCb to allow understanding of its experimental conditions and its performance. It integrates two independent phases (figure2):

- A Event Generator Phase consisting of the generation of p-p collision, the specialized decay and pile-up

- A Detector Simulation Phase consisting in the tracking of the particle through material and simulation the physics processes occurring in the experimental setup. Gauss spends most of the time in this part.

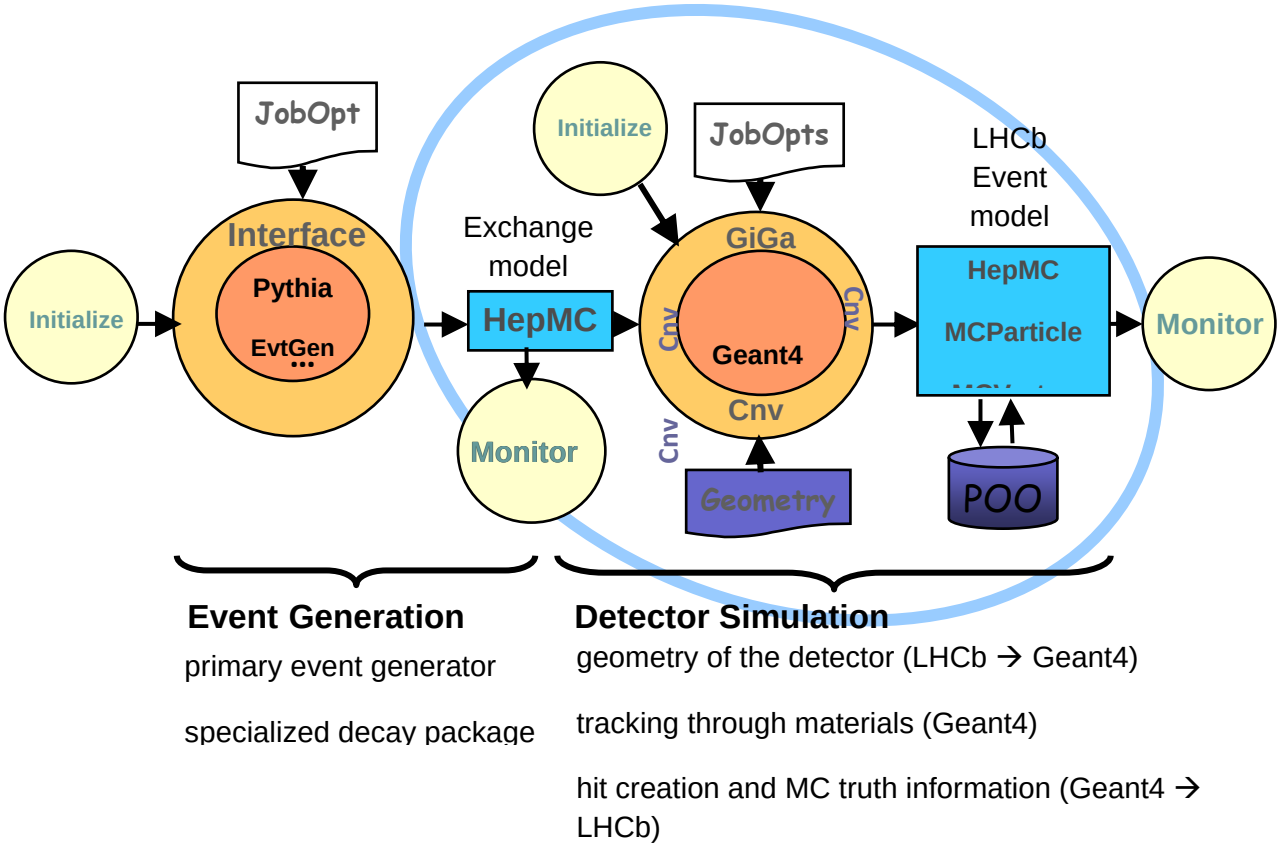


Figure2: Structure of Gauss application

3. Geant4:

Gauss uses Geant4 to simulate LHCb the detector. Geant4 is a toolkit to track particles through matter, transports them and simulates the physics processes. Geant4 comes from mixture of theory-driven, parameterized and empirical formulae. Each particle is moved in steps (from micron to cm) and for each step Geant4 gives information like: material the particle is in, energy loss, directional change due to multiple scattering, effect of electric and magnetic fields on charged particles. For each particle, Geant4 gives information also when it starts and it ends for example: type of the particle, its momentum, and its originating process.

4. Simulation control:

To control the simulation, Geant4 provides production thresholds for electromagnetic processes. It allows us not only to specify range (converted to energy for each material) at which continuous loss of energy begins but also to create secondary particles only above specified range. This is shown in the figure3.

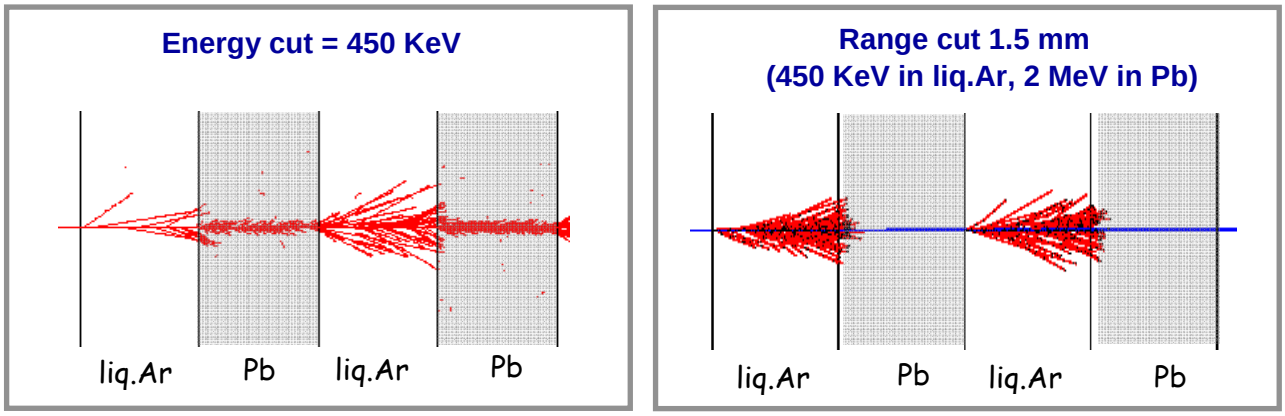


Figure3: Energy cut and range cut

Moreover, one can introduce tracking cut on kinetic energy of particles of all type and for that, Geant4 tracks particle until cut-off energy is reached, stops it at that point. This has been done in Gauss.

Note that it is possible also to set different cuts per volume and region.

5. Monitoring:

The attention of everyone who runs it is drawn that running a job in Gauss is quite slow. It may take several minutes event. Generating enough events requires always a big amount of time. So, the simulation in term of CPU time consumption needs to be kept under control and if possible improved without compromising the results. Thus, the aim of this project is to monitor simulation's performance in term of CPU time described above.

Monitors have been put in place to evaluate this performance in term of CPU time consumption. A code called "MonitorTrackAction" has been created which takes information from Geant4 at the beginning and at the end of track. It gives in the output files:

- Number of the particle found with their names and the tracking time, both as tables and histograms
- Number of the processes, type of processes and time spent for each process, both as tables and histograms

IV- Results:

We used PyRoot, a combination of Python and Root to plot histograms. An example of them for default option is shown in figure 3. In this plot, one can see that Geant4 allocated more times for some types of particle like gamma, electron, positron, optical photon, pi-plus, pi-minus. However, the number of these particles is also significant. By dividing the two fist plots, one can obtain the ratio: tracking time divided by the number of the particle for each type.

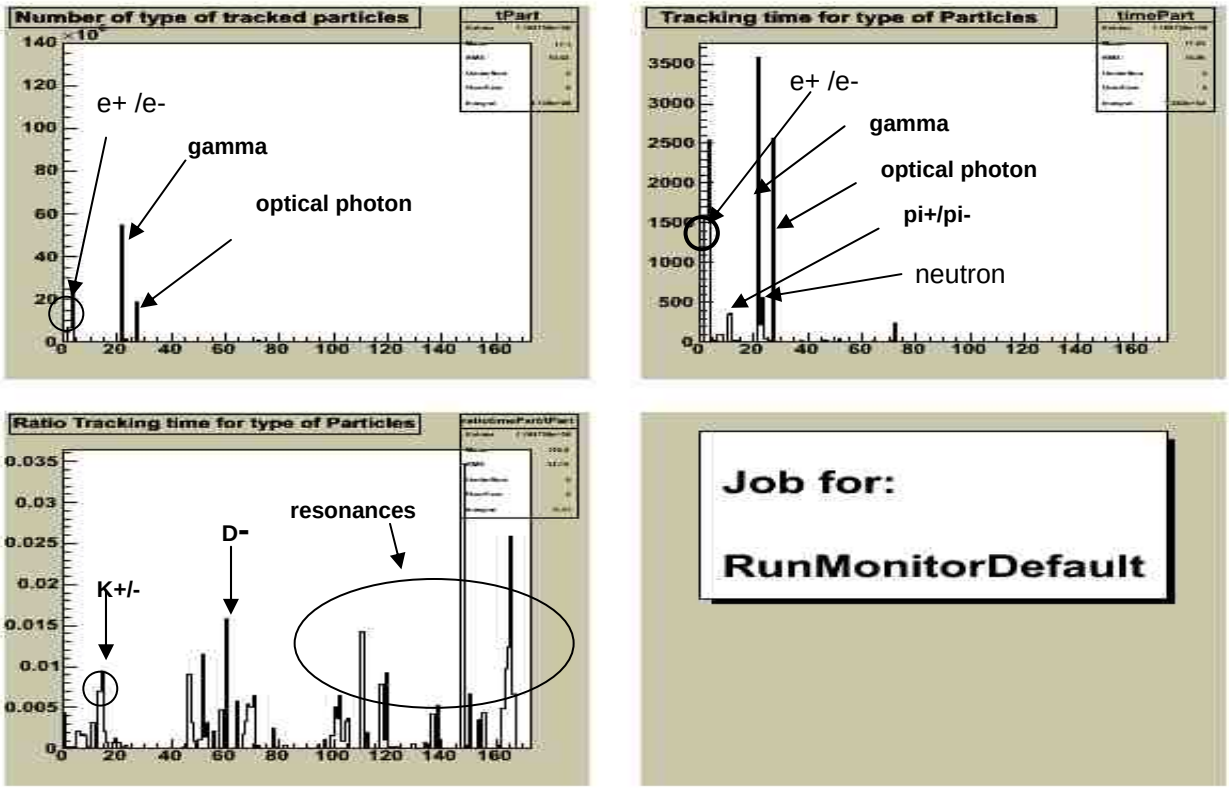


Figure3: Example of histogram for the default option

Effect of cuts on performance:

To study the effect of cuts on performance, four different choices have been used by changing them separately:

- ***Default***, taken as a reference which contains the following options:
  - Production cut: range 10,000 mm for electron, 5mm for positron 10 mm for gamma
  - Tracking cut: 10 MeV for muon, pion, kaon, neutron, 1 MeV for gamma, electron, and zero others

- ***Production cut:*** range was 0.5 mm for electron, positron, gamma
- ***Tracking cut:*** they have been set to zero
- ***Cut Filter:*** 10 MeV for muon+/- and 500 MeV for another particles in the Muon Filter

The results are summarized in the table below in which amounts of time can be compared in each case by regarding just de ratio. One can see that the amount of time higher in the case of ***Production cut*** and even higer for that of the ***Tracking cut***. The number of Geant4 tracks is though significant. The time for the Cut Filter configuration remains almost the same as the default option. One can see particularly that the results in *GiGaFlushAlgorithm Total* and *Total Ttracking inGeant4* are consistent due to the fact that

Geant4 spends most of the time in this algorithm. Note that the total number of Geant4 tracks given in this table is for 500 events minimum bias not a single event. To have the number for one event these values should be divided by 500.

	NCU (s)	**GiGa Flush Average (s)**	**GiGa Flush Min (ms)**	**GiGa Flush Max (s)**	**GiGa Flush Total (s)**	**Total tracking in G4(s)**	**Total Number of G4 tracks**
Default	18691	12.2	7.5	80.2	6160	6193	110875941
		24.9	15.3	163.6	12567	12634	
Ratio	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Prod Cut	35629	23.4	11.2	126.5	11704	11681	165827349
		37.7	18.0	203.7	18843	18807	
Ratio	**1.91**	**1.51**	**1.2**	**1.24**	**1.50**	**1.49**	**1.50**
Track Cut	37000	24.3	7.5	154.2	12167	12191	225830638
		44.9	13.9	285.2	22509	22554	
Ratio	**1.98**	**1.81**	**0.9**	**1.74**	**1.79**	**1.79**	**2.04**
Cut Filter	15521	29.5	16.6	185.7	14866	14851	108132723
		24.8	13.9	156.0	12488	12474	
Ratio	**0.83**	**1.00**	**0.91**	**0.95**	**0.99**	**0.99**	**0.98**

Table: Results for 500 events minimum bias

V- <u>Conclusion</u>:

To conclude this work, we can underline that the performance of the simulation in term of CPU time consumption could be enhanced by putting appropriate and reasonable cuts depending on what we would like to generate as event. In this work we submitted all jobs in

LxBatch generating the machine scaling factor but the best way to compare the result is running all jobs in the same machine.

Acknowledgments:

Many thanks to:

- Pr Stephan NARISON who organized the HEP-MAD 07 and accepted my participation during this conference
- Dr Gloria CORTI, my supervisor during the Summer Student Program , July and August 2007 in LHCb experiment at CERN Geneva Switzerland

References

- Technical Design Report: LHCb Reoptimized Detector Design and performance, CERN/LHCC 2003-030, LHCb TDR, 09 September 2003
- Technical Design Report: LHCb Computing , CERN/LHCC 2005-019, LHCb TDR 11, 20 June 2005
- Technical Proposal: LHCb, CERN/LHCC 98-4, LHCC/P4, 20 February 1998